757 Course Project Report
A Design of Proposal Distribution for MCMC Sampling in non-linear inverse problem

Xinyuan Niu

April 26, 2020

Contents

1 Introduction

An inverse problem in engineering fields is a group of problems that we focus on figuring out unknown responses in terms of observed data set which is generated by them. There are lots of instances in multiple subjects, such as reconstructing an image in radiation tomography, determining source location in acoustics, and calculating the density of the Earth from measurements of its gravity field (examples are from Wikipedia). The whole process starts with a fully known mapping $\boldsymbol{f}$ that is operated onto the parameters $\boldsymbol{\theta}$ in order to build data sets. After observing acquired information which is denoted by $\boldsymbol{y}$; while in real cases noisy contributions need to be added into the available data. Quantitatively, we consider Gaussian noise models:

$$\mathbf{y} = \mathbf{f}(\boldsymbol{\theta}) + \boldsymbol{\epsilon}$$

where $\boldsymbol{y} \in \mathbb{R}^m$ is the observed response, $\boldsymbol{\theta} \in \mathbb{R}^n$ is the unknown parameters, $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m, m \geq n$ is a vector mapping and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ is a $\mathbb{R}^m$ Gaussian noise. If $\mathbf{f}$ is a nonlinear mapping, then the whole process of solving $\boldsymbol{\theta}$ is called a nonlinear inverse problem.

Although it's ideal to figure out explicit solutions for an inverse problem, there are many difficulties that we have to face with when we try to do so, such as non-linearity of mapping, over-determination and the existence of Gaussian noise. Base on that, we turn into figuring out the distribution of unknown parameters $\boldsymbol{\theta}$. Without loss of generality, we let $\boldsymbol{\Sigma} = \mathbf{I}$, then $\boldsymbol{y}$ follows such a distribution whose likelihood function is

$$p(\boldsymbol{y}|\boldsymbol{\theta}) \propto \exp(-\frac{1}{2}||\boldsymbol{f}(\boldsymbol{\theta}) - \boldsymbol{y}||^2)$$

Assume $\boldsymbol{\theta}$ follows a standard Gaussian prior, then the posterior will be

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{y}) &\propto p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \\ &\propto \exp(-\frac{1}{2}(||\boldsymbol{f}(\boldsymbol{\theta}) - \boldsymbol{y}||^2 + ||\boldsymbol{\theta} - \boldsymbol{\theta}_0||^2) \\ &\propto \exp(-\frac{1}{2}||\tilde{\boldsymbol{f}}(\boldsymbol{\theta}) - \tilde{\boldsymbol{y}}||^2) \end{aligned}$$

In general, the posterior we care about is $p(\boldsymbol{\theta}|\boldsymbol{y}) \propto \exp(-\frac{1}{2}||\boldsymbol{f}(\boldsymbol{\theta}) - \boldsymbol{y}||^2)$. Starting from here, we can transfer the inverse problem into a MCMC sampling mechanism. Our ultimate goal is to find a proper proposal distribution for generating samples **as long as objective distribution possesses the above expression**.

2 Randomize-Then-Optimize

In this section we'll go over the method from [1], 'Randomize-Then-Optimize' by which we could robustly while efficiently attain candidate sample data.

2.1 Motivation

If $\boldsymbol{f}(\boldsymbol{\theta})$ is a linear mapping, which is to say $\boldsymbol{f}(\boldsymbol{\theta}) = \boldsymbol{A}\boldsymbol{\theta}$, then the posterior distribution will be

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = (2\pi)^{-\frac{m}{2}}|\boldsymbol{A}^T\boldsymbol{A}|^{\frac{1}{2}}\exp(-\frac{1}{2}||\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{y}||^2)$$

and $\boldsymbol{\theta}$ will follow a Gaussian distribution, with mean $(\boldsymbol{A}^T\boldsymbol{A}^{-1})\boldsymbol{A}^T\boldsymbol{y}$ and variance $(\boldsymbol{A}^T\boldsymbol{A})^{-1}$. Or we can equivalently write $\boldsymbol{\theta}$ as

$$\boldsymbol{\theta} = \boldsymbol{F}_{\boldsymbol{A}}^{-1}(\boldsymbol{v}), \text{where } \boldsymbol{v} = \boldsymbol{A}^T(\boldsymbol{y} + \boldsymbol{\epsilon}),\ \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \text{ and } \boldsymbol{F}_{\boldsymbol{A}} = \boldsymbol{A}^T\boldsymbol{f}(\boldsymbol{\theta})$$

Back to the general non-linear cases, we can imitate the linear case by locally linear approximation with first derivative information.

2.2 RTO method

We first solve MAP problem for $p(\boldsymbol{\theta}|\boldsymbol{y})$, we have

$$\bar{\boldsymbol{\theta}} = \arg\min_{\phi} \frac{1}{2}||\boldsymbol{f}(\boldsymbol{\theta}) - \boldsymbol{y}||^2$$

With first order necessary condition for optima and conducting QR-factorization onto Jacobian matrix $\mathbf{J}(\bar{\boldsymbol{\theta}})$,

$$\bar{\boldsymbol{Q}}^T(\boldsymbol{y} - \boldsymbol{f}(\bar{\boldsymbol{\theta}})) = 0$$

We thus define mapping $\boldsymbol{F}_{\bar{\boldsymbol{\theta}}}(\boldsymbol{\theta}) = \bar{\boldsymbol{Q}}^T\boldsymbol{f}(\boldsymbol{\theta})$ where $\boldsymbol{F}_{\bar{\boldsymbol{\theta}}}(\bar{\boldsymbol{\theta}}) = \bar{\boldsymbol{Q}}^T\boldsymbol{y}$, hence we have $\bar{\boldsymbol{\theta}} = \boldsymbol{F}_{\bar{\boldsymbol{\theta}}}^{-1}(\bar{\boldsymbol{Q}}^T\boldsymbol{y})$. Since $\boldsymbol{f}(\boldsymbol{\theta})$ is continuously differentiable on $\boldsymbol{\theta}$, we have $\boldsymbol{F}_{\bar{\boldsymbol{\theta}}}$ is invertible in a neighborhood of $\bar{\boldsymbol{Q}}^T\boldsymbol{y}$. Motivated by linear case, we can use such a random variable as the proposal distribution to generate candidate samples in MCMC:

$$\boldsymbol{\theta} = \boldsymbol{F}_{\bar{\boldsymbol{\theta}}}^{-1}(\boldsymbol{v}) \text{ where } \boldsymbol{v} = \bar{\boldsymbol{Q}}^T(\boldsymbol{y} + \boldsymbol{\epsilon})|\bar{\boldsymbol{Q}}^T(\boldsymbol{y} + \boldsymbol{\epsilon}) \in \mathrm{ran}(\boldsymbol{F}_{\bar{\boldsymbol{\theta}}}),\ \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I})$$

Thus $\boldsymbol{v}$ is a truncated Gaussian random vector. As for the proposal distribution, Using random variable transformation, we have

$$\begin{aligned} p_{\bar{\boldsymbol{\theta}}}(\boldsymbol{\theta}) &\propto |\bar{\boldsymbol{Q}}^T\boldsymbol{J}(\boldsymbol{\theta})| \times \chi_{\mathcal{R}}(\boldsymbol{F}_{\bar{\boldsymbol{\theta}}}(\boldsymbol{\theta})) \times p_{\boldsymbol{v}}(\boldsymbol{F}_{\bar{\boldsymbol{\theta}}}(\boldsymbol{\theta})) \\ &\propto |\bar{\boldsymbol{Q}}^T\boldsymbol{J}(\boldsymbol{\theta})|\exp(-\frac{1}{2}||\bar{\boldsymbol{Q}}^T(\boldsymbol{f}(\boldsymbol{\theta}) - \boldsymbol{y})||^2) \end{aligned}$$

where $\mathcal{R}$ is the range of $\boldsymbol{F}_{\bar{\boldsymbol{\theta}}}(\boldsymbol{\theta})$ and $p_{\boldsymbol{v}}(\boldsymbol{v}) \propto \exp(-\frac{1}{2}||\boldsymbol{v} - \bar{\boldsymbol{Q}}^T\boldsymbol{y}||^2)$.
What's more, [1] also states that random draws from proposal distribution is equivalent to solving stochastic optimization

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\phi}} ||l(\boldsymbol{\phi}, \boldsymbol{\epsilon}) \triangleq \bar{\mathbf{Q}}^T(f(\boldsymbol{\phi}) - (\boldsymbol{y} + \boldsymbol{\epsilon}))||^2, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

with selecting those $\boldsymbol{\theta}$ such that $l(\boldsymbol{\theta}, \boldsymbol{\epsilon})$ is zero (or small enough e.g. $l(\boldsymbol{\theta}, \boldsymbol{\epsilon}) < \eta = 10^{-8}$ in practice) since the argument that $l(\boldsymbol{\theta}, \boldsymbol{\epsilon})$ is significantly not equal to zero indicates that $\bar{\boldsymbol{Q}}^T(\boldsymbol{y} + \boldsymbol{\epsilon})$ is not in $\mathcal{R}$, i.e. probability density $p_{\bar{\boldsymbol{\theta}}}(\boldsymbol{\theta}) = 0$. Now that we've been fully aware our proposal distribution for candidates, we can embed it into Metropolis-Hastings Algorithm and implement MCMC samplings.

3 Experiments

We will verify the feasibility of RTO method by simulating three examples, two algebraic models: MONOD and BOD, and an atmospheric remote sensing problem with relatively higher dimensions. The outline is, that we first generate $\boldsymbol{y}$ by adding independent and identically distributed Gaussian noise to the curve simulated using the "true" but unknown parameter values, then we reconstruct samples of $\tilde{\boldsymbol{\theta}}$ by Metropolis-Hastings Algorithm using RTO.

3.1 MONOD model

The MONOD model is given by

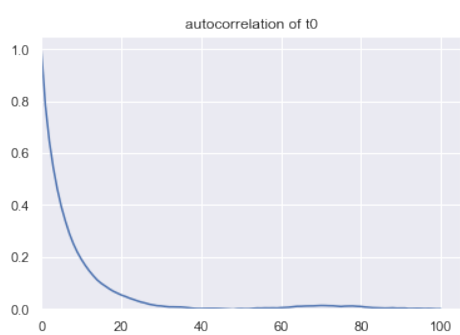
$$\mathbf{f}(\boldsymbol{t}) = \{f_1(\boldsymbol{t}), \ldots, f_m(\boldsymbol{t})\}, \text{ where } f_i(\boldsymbol{t}) = \frac{t_0 x_i}{t_1 + x_i}, i = 1 \ldots m$$

Here $\boldsymbol{x} \in \mathbb{R}^m$ is known parameter vector while $\boldsymbol{t} \in \mathbb{R}^n$ is unknown variable. The factor values and corresponding measurements are given by

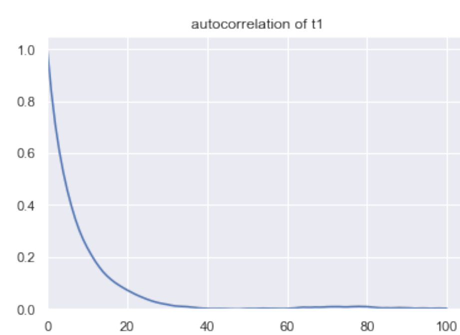
$$\boldsymbol{x} = [28, 55, 83, 110, 138, 225, 375]^T$$

$$\boldsymbol{y} = [0.067, 0.073, 0.101, 0.118, 0.105, 0.120, 0.124]^T$$

Figure 1 indicates that auto-correlation functions for both variables drop down to 0 after a warm-up period and M-H algorithm does generate independent samples. Figure 2 and Figure 3 display the marginal densities and joint densities respectively, while Figure 4 is 3-D probability density captured from two different angles.

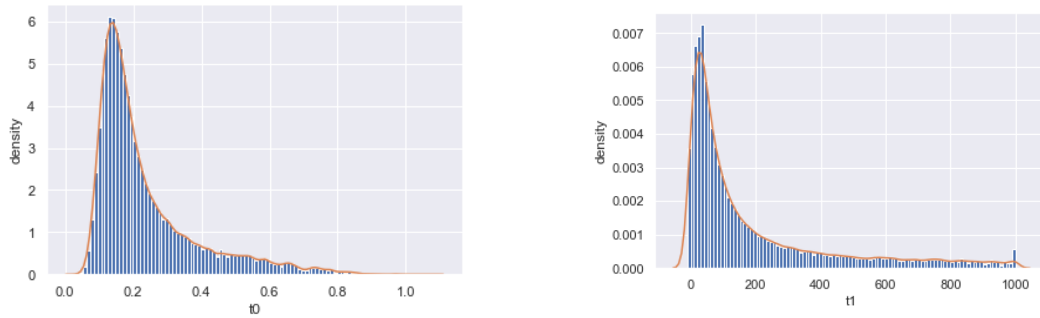


(a) ACF plot of t_0



(b) ACF plot of t_1

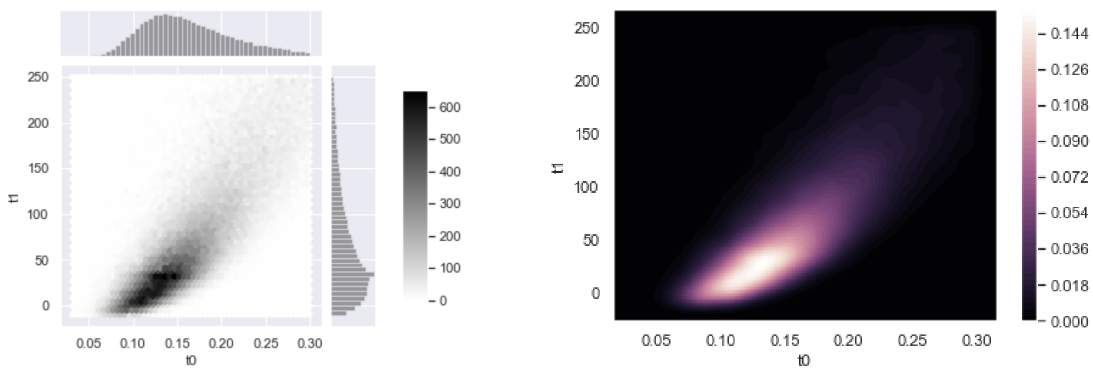
Figure 1: auto-correlation functions for MONOD



(a) marginal density of t_0

(b) marginal density of t_1

Figure 2: marginal density plots based on sample set for MONOD



(a) joint 2-D hex histogram plot

(b) joint probability density plot

Figure 3: joint distribution based on sample set for MONOD

3.2 BOD model

The BOD model is given by

$$\mathbf{f}(\boldsymbol{t}) = \{f_1(\boldsymbol{t}), \ldots, f_m(\boldsymbol{t})\}, \text{ where } f_i(\boldsymbol{t}) = t_0(1 - e^{-t_1 x_i}), i = 1 \ldots m, i = 1 \ldots m$$

Here $\boldsymbol{x} \in \mathbb{R}^m$ is known parameter vector while $\boldsymbol{t} \in \mathbb{R}^n$ is unknown variable. The factor values and corresponding measurements are given by

$$\boldsymbol{x} = [2, 4, 6, 8, 10]^T$$

$$\boldsymbol{y} = [0.1522071, 0.29667172, 0.41254479, 0.48237946, 0.56707723]^T$$

Figure 5 indicates that auto-correlation functions for both variables drop down to 0 after a warm-up period and M-H algorithm does generate independent samples. Figure 6 and Figure 7 display the marginal densities and joint densities respectively, while Figure 8 is 3-D probability density captured from two different angles.

3.3 Atmospheric Remote Sensing: GOMOS

In this subsection We turn into solving an inverse problem in a real case, which is about concentrations of various gases in the atmosphere using measurements collected by the Global Ozone Monitoring

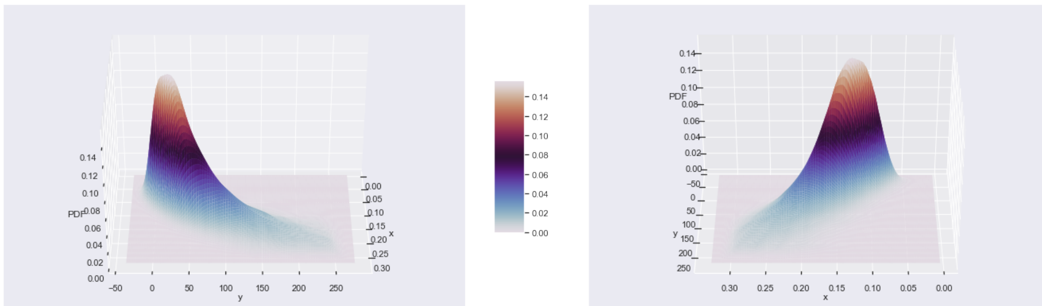
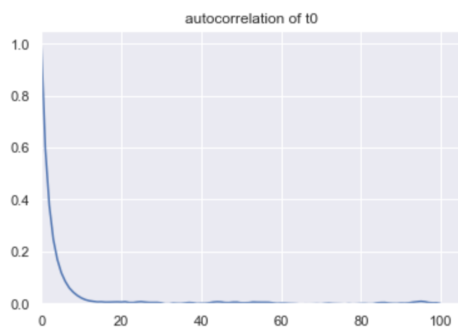
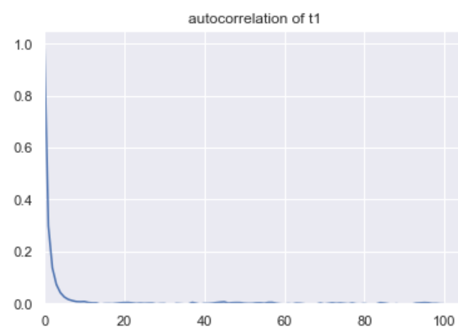


Figure 4: 3-D probability density function plots for MONOD



(a) ACF plot of t_0



(b) ACF plot of t_1

Figure 5: auto-correlation functions for BOD

by Occultation of Stars (GOMOS) instrument. Although the background knowledge of the whole collection process(which can be found in [2] in details) is more than complicate, we'll simplify the reconstruction procedure and formulate it with mathematical model without explaining its physical indication. Essentially the non-linear mapping in this model is

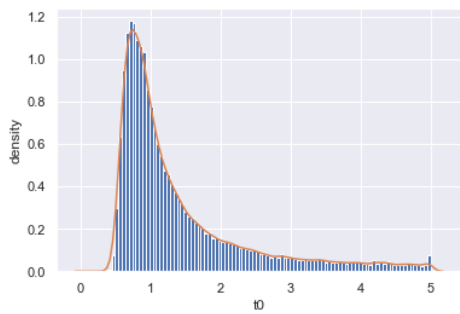
$$\boldsymbol{f}(\boldsymbol{t}) = e^{-\boldsymbol{B}\boldsymbol{t}}$$

where $\boldsymbol{B} \in \mathbb{R}^{m\times m}$ are given matrix generated randomly and $t \in \mathbb{R}^m$ is the unknown parameter vector that we'll sample for. $\boldsymbol{f}$ is obtained by calculated the exponential values component-wisely. Parameters we use here are

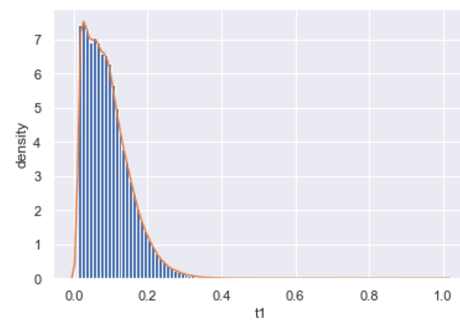
$$\boldsymbol{B} = \begin{bmatrix} 0.141 & 0.979 & 0.002 & 0.182 & 0.496 & 0.582 & 0.259 & 0.681 & 0.639 \\ 0.262 & 0.646 & 0.618 & 0.086 & 0.616 & 0.378 & 0.781 & 0.855 & 0.177 \\ 0.934 & 0.589 & 0.418 & 0.925 & 0.542 & 0.154 & 0.33 & 0.42 & 0.343 \\ 0.937 & 0.853 & 0.254 & 0.223 & 0.408 & 0.399 & 0.327 & 0.606 & 0.708 \\ 0.703 & 0.923 & 0.618 & 0.482 & 0.899 & 0.891 & 0.446 & 0.001 & 0.982 \\ 0.808 & 0.274 & 0.813 & 0.194 & 0.853 & 0.209 & 0.088 & 0.978 & 0.633 \\ 0.051 & 0.385 & 0.78 & 0.386 & 0.237 & 0.269 & 0.975 & 0.999 & 0.977 \\ 0.578 & 0.935 & 0.81 & 0.232 & 0.056 & 0.525 & 0.187 & 0.83 & 0.718 \\ 0.049 & 0.69 & 0.72 & 0.989 & 0.316 & 0.153 & 0.554 & 0.428 & 0.934 \end{bmatrix}$$

$$\boldsymbol{y} = [0.989, 0.988, 1.001, 0.992, 1.005, 0.997, 0.982, 1.004, 0.991]^T$$

Figure 9 shows the data visualization for the 9-dimension sample data set. Specifically, the diagonal terms are the marginal densities for every single component; the non-diagonal terms are the joint probability densities according to its corresponding column and row.

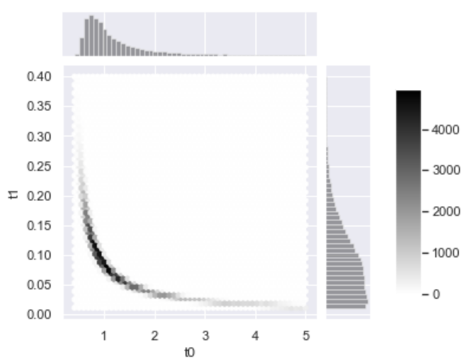


(a) marginal density of t_0

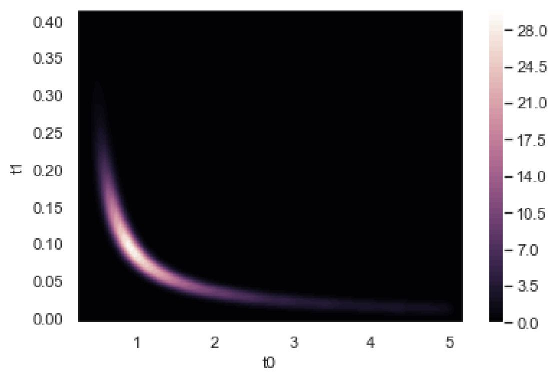


(b) marginal density of t_1

Figure 6: marginal density plots based on sample set for BOD



(a) joint 2-D hex histogram plot



(b) joint probability density plot

Figure 7: joint distribution based on sample set for BOD

References

[1] Johnathan M Bardsley, Antti Solonen, Heikki Haario, and Marko Laine. Randomize-then-optimize: A method for sampling from posterior distributions in nonlinear inverse problems. *SIAM Journal on Scientific Computing*, 36(4):A1895–A1910, 2014.

[2] Heikki Haario, Marko Laine, Markku Lehtinen, Eero Saksman, and Johanna Tamminen. Markov chain monte carlo methods for high dimensional inversion in remote sensing. *Journal of the Royal Statistical Society: series B (statistical methodology)*, 66(3):591–607, 2004.

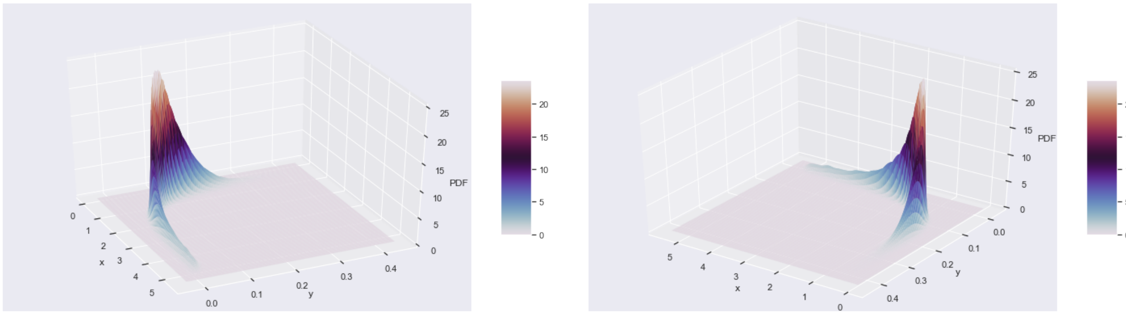


Figure 8: 3-D probability density function plots for BOD

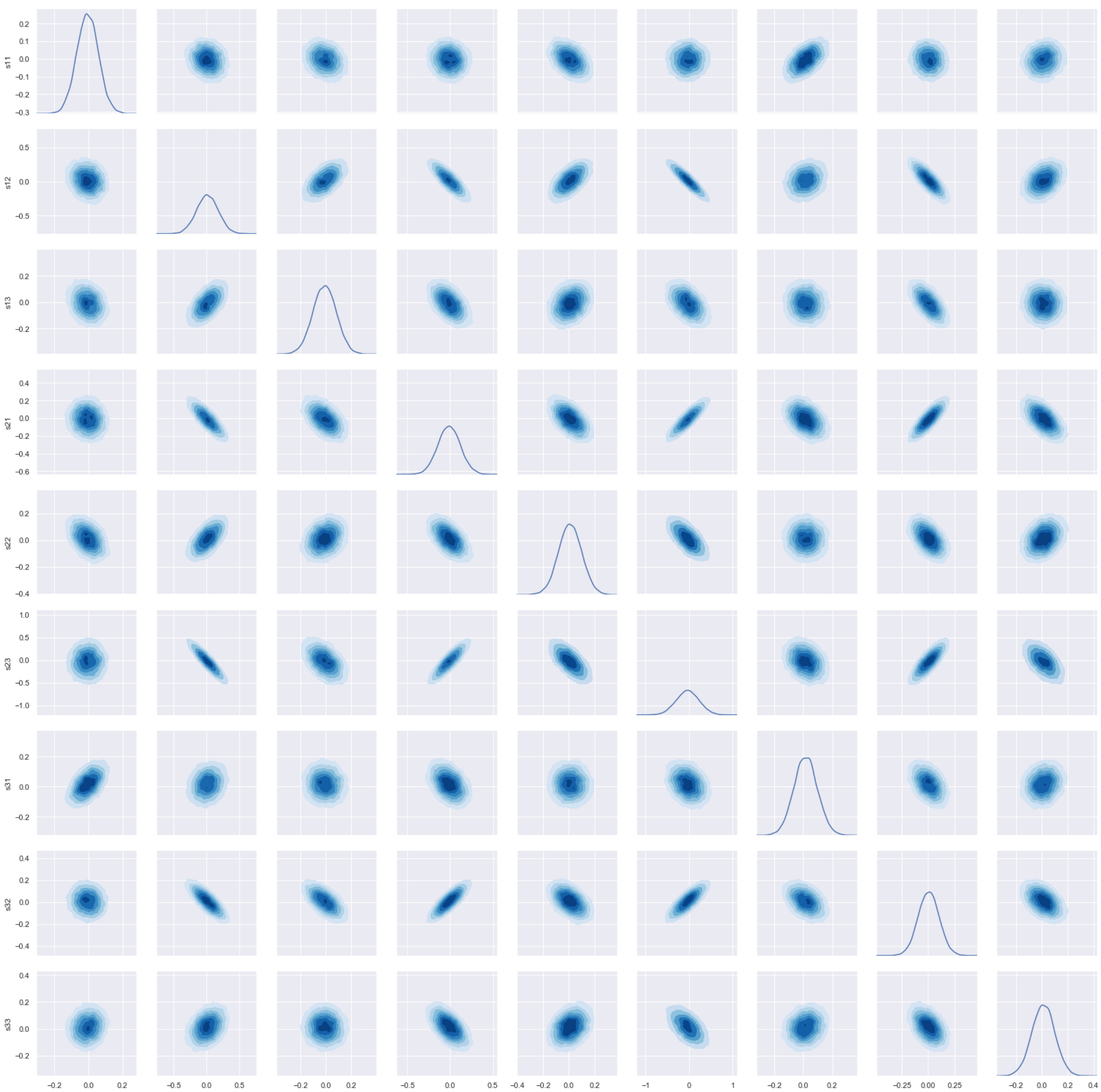


Figure 9: sample data visualization for GOMOS