University of Illinois at Urbana Champaign

A Lovely Yogurt Container

Visualization of the Landscape of Log-likelihood Function of PH Distributions

Nonlinear Programming - IE510

Zhaoxuan Hins Hu

Aug 9, 2020

Contents

1. Introduction

1.1 Background

In the field of statistics and data science, parameter estimation is the pillar for statistical inference, which serves as a key connection between theory and real-world applications. The core idea is to estimate a parameter vector $\mathbf{\Theta}$ of a target random variable X given a set of observed data $\mathcal{D}$ (i.e., realizations of the random variable). Out of many estimation frameworks, maximum likelihood estimation (MLE)[1] and maximum a posterior estimation (MAP) are two generic and popular approaches, which are essentially optimization problems. Some of them are easy unconstrained convex problems (e.g., The MLE for Gaussian random variable can derive the analytical first-order condition), while some of them are high-dimension constrained non-convex problems, in which we can hardly solve the (nearly) optimal parameters even though we can explicitly write down the likelihood function. The focus of this project, which is the MLE of phase-type distributions (PH distributions)is the hard one.

PH distributions play a very important role in many areas like queuing theory and loss modeling because it has many math-friendly closure properties like closure under finite convolution, but from the inference point of view, the setting turns out to be very hard since it involves the estimation of a continuous-time Markov chain (CTMC) of (probably) high dimensions. Some work [1] [2] [3] has introduced algorithms to tackle this problem, which includes the Expectation-Maximization (EM) algorithm and the Markov Chain Monte Carlos (MCMC) scheme.

1.2 Motivations and Goals

The framework and algorithms mentioned above tend to focus more on the statistic world (e.g., stochastic process and MC method) and lacks elaborations from an optimization point of view. Therefore, this project tries to dig deeper into the MLE problem of PH distributions as an optimizer by using the projection visualization technique. Also, the result of visualization may serve as enlightenment to adapt the current Gibbs sampling algorithm to estimate PH distributions of more strict constraints on their parameters.

1. Visualize the landscape of log-likelihood function of PH distribution by choosing a reasonable projection.

2. Discuss some important findings based on the visualized plots.

[1] All abbreviations of professional terms will solely occur in later sections

2. Problem Descriptions

2.1 Phase-Type Distribution

PH distribution is a general type of probability distribution corresponding to random variables that describe the times until the absorption of a Markov process with only one absorbing state. It can be continuous or discrete, depending on whether the stochastic process is a Markov chain or a CTMC. Generally, it has two parameters: 1) The initial probability vector $(\alpha_0, \boldsymbol{\alpha}) \in \mathbb{R}^{n+1}$ indicating the starting state 2) The transition probability/rate matrix $Q \in \mathbb{R}^{n+1}$ of the underlying Markov process, which can be represented by the following form:

$$X \sim PH(\boldsymbol{a}, T)$$

$$Q = \begin{bmatrix} 0 & \mathbf{0} \\ T^0 & T \end{bmatrix}$$

where $\alpha_0 + \boldsymbol{\alpha}\mathbf{1} = 1$, $T^0 + T\mathbf{1} = 0$ and n represents the total number of states excluding the absorbing one.The formal definition of PH distributions can be found in A.1 and A.2, and a simple example of PH distributions can be found in A.3.

2.2 The MLE Problem of PH Distributions

Based on the definition of the probability density function of PH distributions, the likelihood function under the data set of size m can be expressed as follows

$$\mathcal{L}(\boldsymbol{\alpha}, T | X_1 = x_1, \ldots, X_m = x_m) = \prod_{i=1}^{m} \boldsymbol{\alpha} \exp(Tx_i) T_0$$

Now, consider the generic constraints induced by PH distributions, the MLE optimization problem can be completely defined as follow

$$\max_{\alpha, T} \mathcal{L}(\boldsymbol{\alpha}, T | X_1 = x_1, \ldots, X_m = x_m) = \prod_{i=1}^{m} \boldsymbol{\alpha} \exp(Tx_i) T_0$$

$$s.t. \qquad \boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$$

$$T = \begin{bmatrix} -t_{11} & t_{12} & \ldots & t_{1n} \\ t_{21} & -t_{22} & \ldots & t_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1} & t_{n2} & \ldots & -t_{nn} \end{bmatrix} \qquad T^0 = \begin{bmatrix} t_{10} \\ t_{20} \\ \vdots \\ t_{n0} \end{bmatrix}$$

$$\sum_{i=0}^{n} \alpha_i = 1 \qquad \sum_{j=0}^{n} t_{ij} = 0,\ \forall\, i = 1, \ldots, n$$

$$\alpha_i,\ t_{ij} \geq 0,\ \forall\, i = 1, \ldots, n,\ \forall\, j = 0, \ldots, n$$

If we look carefully into the form above, we can observe:

1. The constraints are actually linear constraints in terms of all entries, which can be generally captured by the a large linear system $B\boldsymbol{x} = \boldsymbol{b}$. It indicates that the projected constraints in the 3-D visualization space should be also linear and probably a polygon.

2. All variables are not coupled together. Instead, they can be separated into $n+1$ clusters, namely $(\alpha_0, \ldots, \alpha_n)$, $(t_{10}, \ldots, t_{1n})$, ..., $(t_{n1}, \ldots, t_{nn})$. Linear constraints only exist inside the same cluster, which means we can decouple the whole space of parameters into $n+1$ sub-spaces and treat them separately. It will be discussed in section 5.

3. Visualization of High-Dimension Space

3.1 General Rule for Unconstrained Problems

The main idea of visualization of a function $f : \mathbb{R}^n \to \mathbb{R}$ is to simply project the vector of variables onto a 2-D subspace spanned by columns of a pre-defined $n \times 2$ matrix A, where these two columns serve exactly as the basis of the 2-D space we plan to use. Then, we compute the function values as follows and plot them in the third direction.

$$h(u, v) = f(\boldsymbol{\theta}^0 + u \cdot \frac{A_{\cdot 1}}{\|A_{\cdot 1}\|} + v \cdot \frac{A_{\cdot 2}}{\|A_{\cdot 2}\|})$$

where $\boldsymbol{\theta}^0 \in \mathbb{R}^n$ is the arbitrary origin we choose, $u, v \in \mathbb{R}$ are scalars in two directions for grid (precision) design, and $\| \cdot \|$ refers to vector norms. This method was used to visualize some common neural networks in [5] and was used to explore the trajectories of different minimization methods in [4].

However, the visualization of high-dimension problems are still challenging since the selection of the matrix A (i.e., the projection subspace) is non-trivial, especially when variables in the mathematical expression of f are highly coupled, or pairwise visualizations cannot reveal special structures of variables. Therefore, many techniques are introduced, including the popular one called principle components analysis (PCA) and its variant called kernelized PCA [7]. We will see later that these advanced methods are actually unnecessary or can be used on a much smaller scale since we have a natural way to select a relatively reasonable subspace due to the separable linear constraints.

3.2 PH Cases with Separable Linear Constraints

Based on the constraints formulation in section 2.2, we can conclude the separable linear constraints (neglect the non-negative ones at this moment) by a matrix S as follows, where each row of S represents the normal vector of the hyperplane defined by the corresponding constraint.

$$S_{(n+1)\times(n+1)} = \left[\begin{array}{cccc|c} 1 & 1 & \dots & 1 & 1 \\ \hline -1 & 1 & \dots & 1 & 1 \\ 1 & -1 & \dots & 1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & 1 & \dots & -1 & 1 \end{array}\right] \xrightarrow{\textit{corresponds to}} \left[\begin{array}{cccc|c} \alpha_1 & \alpha_2 & \dots & \alpha_n & \alpha_0 \\ \hline t_{11} & t_{12} & \dots & t_{1n} & t_{10} \\ t_{21} & t_{22} & \dots & t_{2n} & t_{20} \\ \vdots & \vdots & \ddots & \vdots & \\ t_{n1} & t_{n2} & \dots & t_{nn} & t_{n0} \end{array}\right]$$

Then, next step is to randomly and separately pick $n+1$ basis orthogonal to the corresponding row vector, then combine these basis together to form our matrix A.

In fact, if readers are familiar with PH distributions, we can make a smarter decision. Due to the non-uniqueness property of PH representation [6], most of variables are interchangeable in terms of their contributions to the final distribution. Particularly, all non-diagonal entries in the matrix T should have comparable scale in visualization, and all entries in $\boldsymbol{\alpha}$ also do the same. Therefore, in 2-nd to the last row of the matrix S, we can choose the first direction along with t_{ii}, $\forall\ i = 1, \dots, n$, and choose the second direction along with the equal sum of directions of all other entries in the same row. Similarly, we choose the first direction along α_0 and the second direction along with equal sum of directions of all entries in $\boldsymbol{\alpha}$.

Then, the last step is to integrate the non-negative constraints, which is trivial. We only need to check whether every parameter in the original are negative when we generate new (u_0, v_0) in the grid. If none of them are negative, we proceed to compute the log-likelihood at that point, otherwise we discard (u_0, v_0) and replace it with the filler. [1]

By the tricks mentioned above, we can (nearly) visualize the core structure of the high-dimension log-likelihood function of PH distribution with minimal (nearly) information loss.

4. Numerical Experiments

4.1 Example Settings

We use the data set simulated exactly from PH distributions to fit into the likelihood function, such that we can know in advance where global optimum is and set the origin $\boldsymbol{\theta}^0$ as the global optimum for convenience. The following simple PH distribution with 4 states and totally 25

[1] Filler can be the maximum function value we have obtained so far, such that the final 3-D plot will be like a funnel

parameters is considered due to the limitation of computational power.

$$X \sim PH(\boldsymbol{\alpha}, T) \qquad (\alpha_0, \boldsymbol{\alpha}) = (0.2, 0.2, 0.2, 0.2, 0.2)$$

$$T = \begin{bmatrix} -0.2 & 0.05 & 0.05 & 0.05 \\ 0.05 & -0.2 & 0.05 & 0.05 \\ 0.05 & 0.05 & -0.2 & 0.05 \\ 0.05 & 0.05 & 0.05 & -0.2 \end{bmatrix} \qquad T^0 = \begin{bmatrix} 0.05 \\ 0.05 \\ 0.05 \\ 0.05 \end{bmatrix}$$

We first simulate 500 realizations from X. Then, according to section 3, we can pick the projected subspace [1] (i.e., the matrix A) as follows, where ⊠ refers to concatenation of row vectors.

$$\begin{array}{ll} A_{1\cdot}^{\top} = (1, -1, 0.5, 0.5, -1) & A_{2\cdot}^{\top} = (-3/2, -1/4, 1, 1, -1/4) \\ \quad \boxtimes\ (-1, 1, 0.5, 0.5, -1) & \quad \boxtimes\ (-1, -3/2, 1, 1, 1/2) \\ \quad \boxtimes\ (1, -1, 0.5, 0.5, -1) & \quad \boxtimes\ (-3/2, -1, 1, 1, 1/2) \\ \quad \boxtimes\ (0.5, 1, -1, 0.5, -1) & \quad \boxtimes\ (1, -3/2, -1, 1, 1/2) \\ \quad \boxtimes\ (0.5, 1, 0.5, -1, -1) & \quad \boxtimes\ (1, -3/2, 1, -1, 1/2) \end{array}$$

4.2 Results of Visualization

By the help of package `actuar` in R community and the visualization library `plotly`, we successfully visualize the landscape of the log-likelihood function of PH distribution, whose lovely shape is like a yogurt container.[2] Notice that the yellow flat upper cap represents the infeasible region.

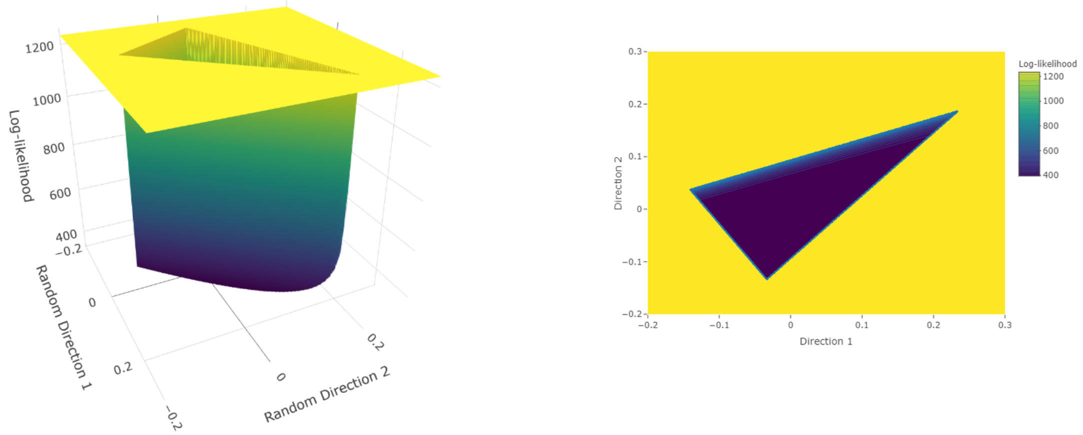

Figure 4.1: Visualization of Log-likelihood Landscape of PH Distributions

[1] Different random directions are also picked to generate different plots, but they are almost the same in appearance. See B.1 for more details

[2] Two different viewing angles can be found in B.2. Also, interactive visualization can be found in https://github.com/Hins1996/Optim-Visualization

5. Discussion of Interesting Findings

5.1 Convex or Non-convex?

Recall the original likelihood function is $\mathcal{L}(\boldsymbol{\alpha}, T|X_1 = x_1, \ldots, X_m = x_m) = \prod_{i=1}^{m} \boldsymbol{\alpha} \exp(Tx_i)T_0$, we may naturally think the problem is non-convex at the very beginning because it involves a matrix exponential where every element of the matrix is a variable, implying very high-level couple effects between variables. However, after visualizing the projected log-likelihood [1], we find that the projected function in the constrained set is very likely convex because we cannot observe any concavity in the plots (Recall that in [5] we can still observe concavity of the projected loss landscape of neural nets even though the plots are smooth enough).

Another observation is that: even though the projected likelihood is convex, the original likelihood function in the high dimension space may still be non-convex. The current result may happen to capture the convexity in the selection of random directions, especially when the constraints in the general PH distributions are quite special.

The final answer is hidden in the mathematics itself, and it needs a deeper search in math literature and even needs a novel proof.

5.2 Polygon Constraints in the Projection Space

As mentioned in section 2, the projection of linear constraints and non-negative constraints are still linear since projection is one type of linear transformation. We may also notice that different choices of random directions may result in different polygons in 2-D space. Based on it, we can ask questions: how is the result of numerical experiments related strict mathematical expression? Is the constrained set always be a polygon? Is it possible to be a unbounded set?

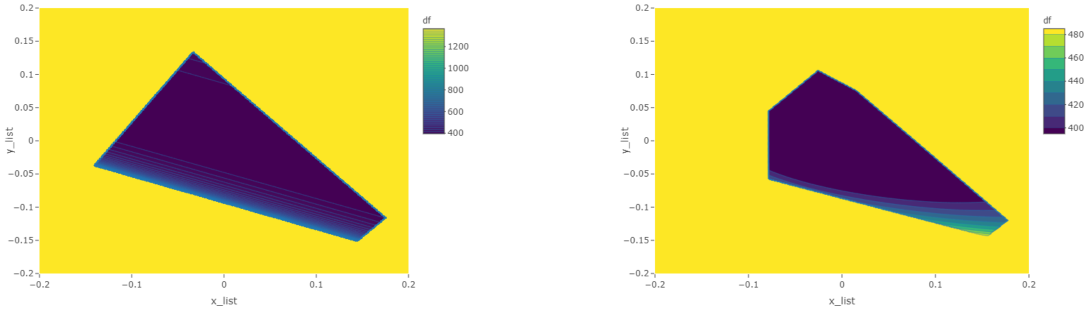

Figure 5.1: Different Polygons in 2-D Space

[1] Log(f(x)) preserve the convexity if f(x) is convex, but indicate nothing if f(x) is non-convex

All of these questions become trivial if we treat the projection as a linear transformation described by the projection matrix $P = A(A^\top A)^{-1}A^\top$, where A is the $n\times 2$ matrix determining the projection subspace. Some hand derivations can direct to the answer.

5.3 Normalization of Random Directions

Normalization of directions is key for projection visualization because the landscape should be invariant to the scale of directions. [4] and [5] also emphasize it. However, we need to be caveat that invariance property implies both directions should be on the same scale, but doesn't imply that different components in the same direction should be on the same scale.

Back to our separable problem defined in section 2, we need to keep each separated cluster of variables on their own scale, otherwise some of them may not contribute enough to the change of landscape when we tune the scalar (u, v). For example, variables in the cluster of (α_0, α) are limited to $[0, 1]$ while all other variables can take $[0, \infty)$. If we normalize all other clusters to the same scale of vector (α_0, α), they will become insensitive to the change of (u, v).

5.4 Flat Bottom and Steep Margin

A key fact is: no matter what random directions we choose or what data set we adopt, the log-likelihood landscape always drops very fast near the margin of constraints and keeps almost the same level when it touches the bottom. That's why it looks like a lovely yogurt container. This so-called effect may cause hardness in optimization if we don't choose algorithms carefully. For example, gradient projection (GP) may fail because the Lipschitz gradient of the likelihood function may not exist. It verifies that only a limited number of tailored algorithms work well in estimating general PH distributions.

6. Conclusion and Feature Work

In this project, we visualize the landscape of log-likelihood function of PH distributions, which captures the core "shape" of the problem we define in section 2 and reveals some interesting facts mentioned in section 5. It enlightens the future work, including:

1. Visualize the optimization trajectory of the MCMC scheme (Gibbs sampling) in estimating general PH distributions in the current plots
2. Adapt the current Gibbs sampling algorithm [2] to estimate a structured PH distribution with more strict constraints.

Bibliography

[1] Søren Asmussen, Olle Nerman, and Marita Olsson. Fitting phase-type distributions via the em algorithm. *Scandinavian Journal of Statistics*, pages 419–441, 1996.

[2] Mogens Bladt, Antonio Gonzalez, and Steffen L Lauritzen. The estimation of phase-type related functionals using markov chain monte carlo methods. *Scandinavian Actuarial Journal*, 2003(4):280–300, 2003.

[3] Luz Judith R Esparza. Maximum likelihood estimation of phase-type distributions. 2011.

[4] Ian J Goodfellow, Oriol Vinyals, and Andrew M Saxe. Qualitatively characterizing neural network optimization problems. *arXiv preprint arXiv:1412.6544*, 2014.

[5] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pages 6389–6399, 2018.

[6] Colm Art O'Cinneide. On non-uniqueness of representations of phase-type distributions. *Communications in Statistics. Stochastic Models*, 5(2):247–259, 1989.

[7] Changshui Zhang, Feiping Nie, and Shiming Xiang. A general kernelization framework for learning algorithms based on kernel pca. *Neurocomputing*, 73(4-6):959–967, 2010.

Appendix A

A.1 Discrete PH Distribution

Consider an underlying discrete-time finite discrete-state homogeneous Markov chain, with states $0, 1, 2, \ldots, n$, where the state 0 is an absorbing state, while the states $1, 2, \ldots, n$ are transient states, or so-called phases. Denote α_0 as the probability that the Markov chain initiates at the absorbing state 0, while $\boldsymbol{\alpha}$ as a (column) vector of size n representing the probabilities that the Markov chain initiates at the respective phases $1, 2, \ldots, n$. Let

$$P = \begin{bmatrix} 1 & \mathbf{0} \\ T^0 & T \end{bmatrix}$$

be the transition probability matrix of the Markov chain, where $\mathbf{0}$ is the row vector of size n containing all zero elements; T^0 and T are respectively a vector, of size n, and a matrix, of size $n \times n$, in which both elements lie in $[0, 1]$, such that $T^0 + T\mathbf{1} = 0$, with $\mathbf{1}$ being the vector of size n containing all unit elements. If K is defined as the (random) number of transitions required to be absorbed to the state 0 for the Markov chain, then K is called an n-phase discrete phase-type distribution with parameters $\boldsymbol{\alpha}$ and T, which is denoted as $\mathrm{PH_d}\left(\boldsymbol{\alpha}, T\right)$. More specifically, the probability mass function and the distribution function of K are given by: $p_K\left(0\right) = F_K\left(0\right) = \alpha_0$, and, for $k = 1, 2, \ldots, n$

$$p_K\left(k\right) = \boldsymbol{\alpha}^\top T^{k-1} T^0;$$

$$F_K\left(k\right) = 1 - \boldsymbol{\alpha}^\top T^k \mathbf{1}.$$

A.2 Continuous PH Distribution

Consider an underlying continuous-time finite discrete-state homogeneous Markov chain, with states $0, 1, 2, \ldots, n$, where the state 0 is an absorbing state, while the states $1, 2, \ldots, n$ are transient states, or so-called phases. Denote α_0 as the probability that the Markov chain initiates at the absorbing state 0, while $\boldsymbol{\alpha}$ as a (column) vector of size n representing the probabilities that the Markov chain initiates at the respective phases $1, 2, \ldots, n$. Let

$$Q = \begin{bmatrix} 0 & \mathbf{0} \\ T^0 & T \end{bmatrix}$$

be the transition rate matrix of the Markov chain, where $\mathbf{0}$ is the vector of size n containing all zero elements; T^0 and T are respectively a vector, of size n, and a matrix, of size $n \times n$, in which all elements except the diagonal ones of T are non-negative and the diagonal ones of T are negative, such that $T^0 + T\mathbf{1} = \mathbf{0}$, with $\mathbf{1}$ being the vector of size n containing all unit elements. If X is defined as the (random) time required to be absorbed to the state 0 for the Markov chain, then X is called an n-phase continuous phase-type distribution with parameters $\boldsymbol{\alpha}$ and T, which is denoted as PH $(\boldsymbol{\alpha}, T)$. More specifically, the probability density function and the distribution function of X are given by: for $x \geq 0$,

$$f_X(x) = \boldsymbol{\alpha}^\top \exp(Tx)\, T^0;$$

$$F_X(x) = 1 - \boldsymbol{\alpha}^\top \exp(Tx)\, \mathbf{1}.$$

where $\exp(Tx)$ refers to the matrix exponential on the matrix Tx

A.3. A Simple Example of PH Distribution

Due to the broad definition, PH distribution is actually a set of many commonly seen distributions, including exponential distribution, gamma distribution, Coxian distribution, and so on so forth. What's more, the convolution and mixture of simple PH distributions are also of phase-type. Now consider $X \sim PH(\boldsymbol{\alpha}, T)$ is a PH distribution with initial probability vector $\boldsymbol{\alpha} = [0, 0.5, 0.5]$ and the transition rate matrix

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ \lambda_1 & -\lambda_1 & 0 \\ \lambda_2 & 0 & -\lambda_2 \end{bmatrix}$$

Obviously, the underlying CTMC has three states with state 0 as the absorbing state. Both state 1 and state 2 can be the initial state with probability 0.5. Essentially, it has different exponential rates of time spent in state 1 and state 2, namely λ_1 and λ_2, which makes it the mixture of two exponential distributions $\exp(\lambda_1)$ and $\exp \lambda_2$, or in other words, a hyper-exponential distribution. Therefore, X refers to how long this CTMC has endured before it's trapped by state 0.

Appendix B

B.1 Different Random Directions of Projection

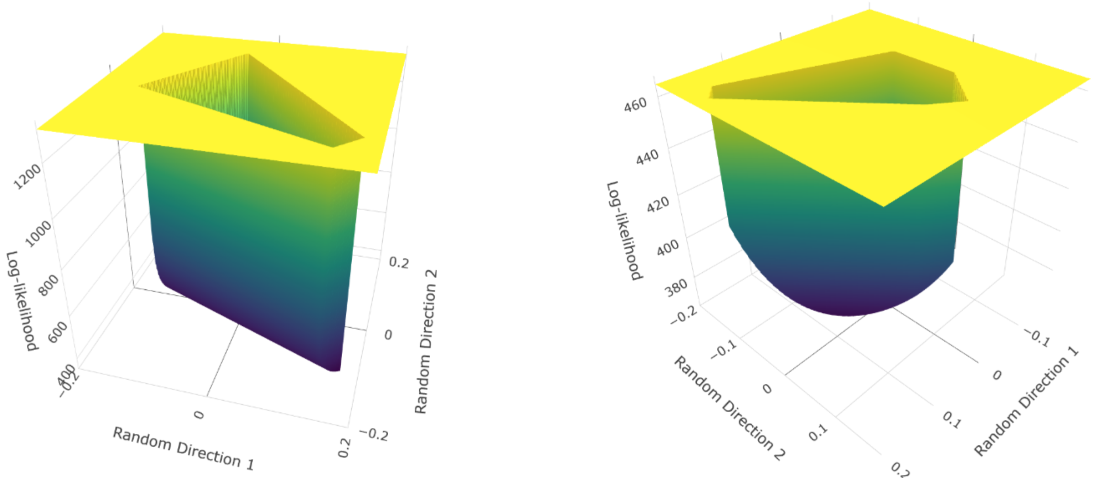

Figure 6.1: Different Random Directions of Projection

B.2 Different Viewing Angle of the Landscape

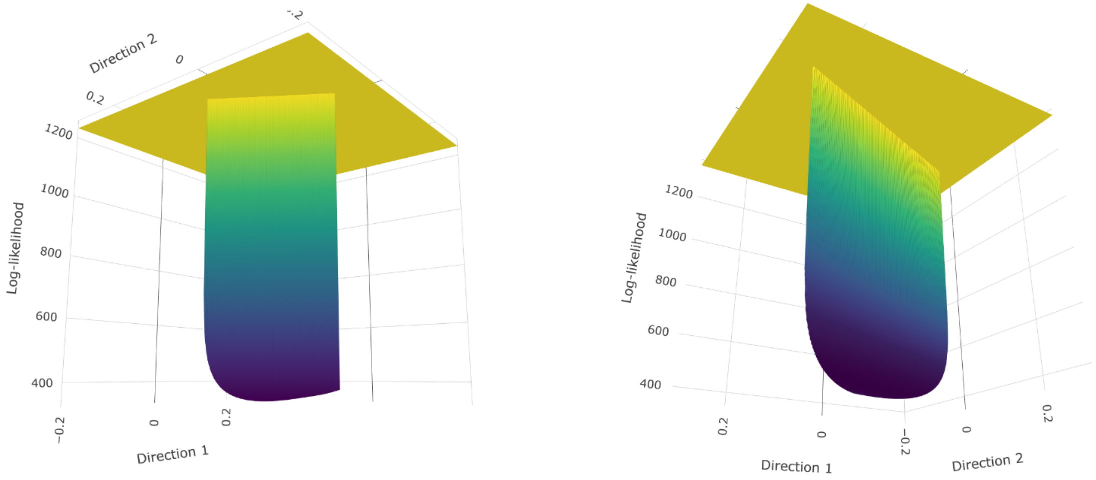

Figure 6.2: Different Views of the Landscape