Nonparametric Bayesian Density Estimation with Gaussian Processes

by

Haoxuan Wang

Department of Statistical Science
Duke University

Defense Date: April 3, 2024

Approved:

Surya Tokdar, Supervisor

Simon Mak

David Banks

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in the Department of Statistical Science
in the Graduate School of Duke University
2024

ABSTRACT

Nonparametric Bayesian Density Estimation with Gaussian Processes

by

Haoxuan Wang

Department of Statistical Science
Duke University

Defense Date: April 3, 2024

Approved:

Surya Tokdar, Supervisor

Simon Mak

David Banks

An abstract of a thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in the Department of Statistical Science in the Graduate School of Duke University
2024

Abstract

This thesis presents a comprehensive study on nonparametric Bayesian density estimation using Gaussian processes (GP). We explore the logistic Gaussian Process (LGP) and introduce an innovative approach termed the tree-logistic-link Gaussian process (TLLGP). This method aims to improve computational efficiency while maintaining modeling flexibility. We address the computational challenges traditionally associated with LGP by implementing a novel tree-based strategy, thereby reducing the complexity of posterior computations. Through a series of numerical experiments, we demonstrate the effectiveness of TLLGP in various scenarios, comparing its performance with other methods. The results highlight the advantages of our approach in terms of computational speed and accuracy in density estimation tasks. This work contributes to the fields of Bayesian statistics and machine learning by providing a more efficient tool for density estimation, especially beneficial for large high-dimensional data where traditional methods fall short due to their computational demands.

Dedication

To my family

Contents

List of Tables

List of Figures

Acknowledgements

First and foremost, I would like to express my sincere appreciation to my thesis advisor, Dr. Surya Tokdar. His deep insights in Gaussian processes have had a profound impact on my work, and I was continually inspired by his dedication to challenging and meaningful problems. Dr. Tokdar not only taught me how to build motivation broadly, formulate ideas rigorously, and analyze problems carefully, but he also instilled in me the values of hard work, good communication, and perseverance. Despite the challenging nature of this project, I never felt discouraged with his support.

Also, I would like to extend my profound gratitude to Dr. David Dunson, who advised me on another project "Bayesian modeling of multi-species labeling errors in ecological studies" within his LIFEPLAN. I deeply admire his immense knowledge and care for his students. Without his support, I would not have made it through the tough semester of my Ph.D. application and ended up with a dream program to go to.

I thank my previous collaborator Minhui Jiang for his help on numerical experiments at the beginning of this research project and inspirational discussions. Thanks also to my another collaborator Patrik Lauha from Finland, helping me quickly pick up the application background of LIFEPLAN.

And I would like to thank Dr. Simon Mak and Dr. David Banks for serving as my committee members. Thanks a lot to Dr. Yuansi Chen, Dr. Li Ma, Dr. Scott Schmidler for their wonderful lectures.

Lastly, I want to express my gratitude to my friends at Duke and from my undergraduate studies. Particularly, words cannot express my appreciation to my parents and grandparents for their companionship and unconditional love throughout my life!

1. Introduction

Gaussian processes (GPs) enable us to specify flexible prior distributions over functions and have been widely used in nonparametric regression and classification due to their computational tractability and asymptotically optimal statistical performance (Van der Vaart & Van Zanten, 2009; Williams & Rasmussen, 2006). Therefore, it's natural to carry out Bayesian density estimation with GPs. Nevertheless, sample paths of a GP might not be non-negative and normalised to 1, which further implies that random functions generated from a GP ought to be passed through a non-linear link function and the corresponding results need noramalisation. And different link functions offer us various choices of density estimation.

1.1 Logistic Gaussian Process

Logistic Gaussian process (LGP) priors for Bayesian density estimation were proposed and studied by (Lenk, 1988, 1991; Leonard, 1978; Tokdar, 2007). We could easily specify the priors over density functions with the mean and the covariance function of the underlying GP. Usually, we choose to use a mean zero stationary GP with the sqaured-exponential covariance function $k(u,v) = \kappa^2 \exp\left(-\lambda^2(u-v)^2\right)$, where the inverse length-scale $\lambda > 0$ determines the range of smoothing.

Let $\omega = \{\omega(t) : t \in T\}$ be a a mean zero stationary GP with the covariance $k(u,v) = \kappa^2 \exp\left(-\lambda^2(u-v)^2\right)$ on a compact set $T \subseteq \mathbb{R}^d$. Then we could define LGP as

$$(L\omega)(t) = \frac{e^{\omega(t)}}{\int_T e^{\omega(s)} ds}, t \in T, \tag{1.1}$$

leading to a continuous probability density function on T. Furthermore, if λ^d is assigned a gamma prior, the corresponding Bayesian method offers asymptotically optimal density estimation by automatically adapting to the unknown smoothness level of any continuous density function f^* with $\|\log f^*\|_\infty < \infty$ (Van der Vaart & Van Zanten, 2009).

In certain situations, a GP based density estimation may be preferred to the use of mixture priors. (Tokdar et al., 2024) show that a semiparametric extension of the LGP

prior offers optimal estimation of both the density as well as the tail index of heavy-tailed distributions, and could be useful for model-based extreme value analysis. Comparable theoretical guarantees have proven elusive to mixture based formulations (Li et al., 2019). Furthermore, relative to mixture priors, LGP priors are simpler to generalize to the regression setting where one is interested in nonparametrically estimating a conditional density function (Tokdar et al., 2010).

Unfortunately, relative to either density estimation with mixture priors or regression and classification with GP priors, density estimation with LGP priors offers far less computational tractability. Posterior computation is quite challenging since the likelihood function depends on all the infinitely many GP values in the domain T through the normalization operation in (1.1) rather than on the finite number of function values. In low-dimensional cases, the integral involved could be efficiently approximated via discretization so that posterior computation needs to handle only a vector of realizations $W = (\omega(\tilde{t}_1), \omega(\tilde{t}_2), \ldots, \omega(\tilde{t}_r))$; see (Tokdar, 2007). However, due to the lack of any partial conjugacy structures, posterior computation mostly relies on random walk based Markov chain Monte Carlo, which could be slow to mix even for moderately large r due to the high correlation among $\{\omega(\tilde{t}_1), \omega(\tilde{t}_2), \ldots, \omega(\tilde{t}_r)\}$. This problem is further aggravated in density regression, where a representational vector W must have $n \times r$ coordinates, where n is the sample size and r denotes the chosen discretization size for the normalization operation.

1.2 Other Bounded Link Functions

As mentioned above, analytical inference is impossible for LGP priors. (Adams et al., 2009) proposed another interesting MCMC sampling algorithm to partially alleviate the computational challenges by adopting a bounded positive link function $\Phi(\cdot)$ rather than the exponential function. The boundness of the link function enables us to apply rejection sampling and instantiate the rejected proposals before each observation. Specifically, we

could model the probability density as

$$f(t) = \frac{\Phi\left(\omega(t)\right)}{\int_T \Phi\left(\omega(s)\right) ds}, \tag{1.2}$$

where the proposal distribution in rejection sampling is the uniform distribution over T. Suppose we have observations $\{y_1, y_2, \ldots, y_n\}$ from some probability density function. If we could recover the rejection history $\left\{t_{i,1}, t_{i,2}, \ldots, t_{i,r_i}\right\}$ preceeding observation y_i, the augmented-likelihood will be

$$L(\omega) = \prod_{i=1}^{n} \Phi(\omega(y_i)) \prod_{j=1}^{r_i} \left[1 - \Phi(\omega(t_{i,j}))\right], \tag{1.3}$$

which doesn't involve the integral of the whole sample path. Moreover, this augmented-likelihood is identical to the likelihood one would compute in a binary classification problem where each rejected proposal has a label $z_{i,j} = 0$ and each observation has a label $z_{i,r_i+1} = 1$. These labels could be modeled as $z_{i,j} \sim \text{Bernoulli}(\Phi(\omega(t_{i,j})))$. For suitable choices of the link function Φ, the binary classification problem could be handled efficiently due to various data augmentation techniques, such as (Albert & Chib, 1993) for probit link and (Polson et al., 2013) for logistic link, etc. These augmented-data Gibbs samplers could be easily extended to recover the rejection histories of observations. To stochastically regenerate the rejection history, we could simply run the rejection sampling algorithm until one proposal is accepted. Afterwards, we can discard the accepted proposal, retain all the rejected proposals, and generate a random draw of the rejection history conditional on ω; see (Adams et al., 2009; Rao et al., 2016) for more details.

Unfortunately, the modeling approach and computational techniques we have mentioned above also face their own practicability issues. In our numerical experiments, we find that rejection histories can vary in length as the sampling algorithm progresses and could get arbitrarily large at any given instance. This variability poses three problems. Firstly, efficient software implementation becomes very challenging, especially on the question of memory allocation. Secondly, large rejection histories can rapidly slow down computational

speed, since updating ω and its covariance parameters requires factorizing or inverting $N \times N$ matrices, where $N = n + r_1 + r_2 + \cdots + r_n$ is the total number of observations and rejected proposals. Such matrix operations have $O(N^3)$ computational complexity, which could be prohibitive for large N. Lastly, as we demonstrate in numerical experiments, the overall Markov chain is slow mixing and sojourns to large values of N are persistent, further increasing the overall runtime of the algorithm.

1.3 An Alternative with Capped Computation

Here we propose an alternative way of using GP priors for density estimation. The key is a new construction of a density function $\eta(t)$ on T from a continuous but otherwise unconstrained function $\omega(t)$. Similar to the model strategy proposed in (Adams et al., 2009), a random sample from our $\eta(t)$ could be drawn by running a sampling algorithm which requires evaluating the original $\omega(t)$ at a finitely many random points drawn uniformly from T.

This sampling algorithm can be characterized by a binary tree representing a tournament bracket. One starts with uniformly drawn candidates at the leaf nodes of the tree. Each candidate t is given a score of $\exp(\omega(t))$. The candidates are stochastically filtered upward in head-to-head contests with winning probabilities proportional to their scores until a single champion rises to the root node. With a GP prior on ω, posterior computation can proceed by a Gibbs sampler which alternates between completing the sampling history given ω and updating ω as well as related parameters given the augmented-data via a modification of (Polson et al., 2013). A key point of departure from (Adams et al., 2009) is that the size of the augmented-data is fixed at $N = n \times 2^m$ where m is the depth of the sampling tree. The fixed size largely alleviates the problems discussed earlier (see more details in Chapter 3).

It should be noted that while this gain in computational complexity is significant, it comes with the trade-off of reduced shape flexibility in the constructed function $\eta(t)$. This limitation is quantified as $\|\eta\|_\infty \leqslant 2^m$, indicating a bounded behavior which might not

be suitable for all types of datasets or applications. Therefore, it becomes imperative to carefully tune the hyperparameter m, treating it as a crucial tuning parameter to balance the trade-off between computational efficiency and model flexibility. In practice, the tuning of m should be approached with consideration of the specific context and requirements of the application. For instance, a lower value of m might suffice for simpler datasets or problems where a rough approximation of the underlying process is acceptable. Conversely, for more complex datasets or where precision is paramount, a higher value of m might be necessary, albeit at the cost of increased computational resources. We show in Chapeter 4 good performances can be attained with a relatively modest value of m. Chapter 3 gives technical details of the new tree-logistic-link Gaussian process (TLLGP) as well as its fundamental mathematical properties.

2. Logistic-link Gaussian Process

This chapter gives technical details of the logistic-link Gaussian process (LLGP).

2.1 Model

Assume we have observed n independent samples $y_1, y_2, \ldots, y_n$ from a data space T, we hope to model the probability density on such space and express our prior belief on probability density functions via GPs. Initially, we place a GP prior over a scalar function $\omega(t) : T \to \mathbb{R}$. The mean and covariance functions are parameterized by hyperparameter θ. In order to model the density over T, we need a map from $\omega(t)$ to a probability density function $f(t)$ via

$$f(t) = \frac{\Phi(\omega(t))\,\pi(t)}{\int_T \Phi(\omega(s))\,\pi(s)ds}, \tag{2.1}$$

where $\pi(t)$ is a base probability density on T and $\Phi(t)$ is a link function mapping $\mathbb{R}$ into $(0,1)$. Then, we denote $\mathcal{Z}_\pi[\omega] = \int_T \Phi(\omega(s))\,\pi(s)ds$. In this case, we'd like to adopt the logistic function as our link function, which possesses nice properties and is appropriate to use rejection sampling to generate samples from the GP.

2.2 Data Augmentation

As we have stated above, we observe n data $\mathcal{D} = \{y_i\}_{i=1}^n$ that we model as being drawn independently from some unknown density $f(t)$. Our prior belief for the probability density $f(t)$ is quantified by the GP. Here we use the notation $\boldsymbol{\omega}$ and $\mathbf{f}$ to represent the function $\omega(t)$ and $f(t)$ separately. By Bayes' Theorem, the posterior of $\boldsymbol{\omega}$ is

$$\begin{aligned} p(\boldsymbol{\omega} \mid \mathcal{D}; \theta) &= \frac{p(\boldsymbol{\omega} \mid \theta) p(\mathcal{D} \mid \boldsymbol{\omega})}{\int p(\boldsymbol{\omega}' \mid \theta) p(\mathcal{D} \mid \boldsymbol{\omega}') d\boldsymbol{\omega}'} \\ &= \frac{p(\boldsymbol{\omega} \mid \theta)\,(\mathcal{Z}_\pi[\omega])^{-n} \prod_{i=1}^n [\Phi(\omega(y_i))\,\pi(y_i)]}{\int p(\boldsymbol{\omega}' \mid \theta)\,(\mathcal{Z}_\pi[\omega]')^{-n} \prod_{i=1}^n [\Phi(\omega'(y_i))\,\pi(y_i)]\,d\boldsymbol{\omega}'}. \end{aligned} \tag{2.2}$$

It's difficult to evaluate the posterior distribution over $\boldsymbol{\omega}$ due to the denominator and the normalization constant $\mathcal{Z}_\pi[\omega]$. Below we propose an MCMC algorithm incorporating

the rejection sampling and introducing the Pólya-Gamma random variable (Polson et al., 2013), which sidesteps this difficulty.

2.2.1 Latent History of Rejection Sampling

In our problem, the sampling model for one observation is $p(t \mid \boldsymbol{\omega}) = \frac{\Phi(\omega(t))\pi(t)}{\int_T \Phi(\omega(s))\pi(s)ds}$, which involves a normalization constant hard to evaluate. Sampling directly from $p(t \mid \boldsymbol{\omega})$ isn't easy. Fortunately, the kernel of $f(t)$ is bounded since we adopt the logistic function as the link and then $\pi(t) \geqslant \Phi(\omega(t))\,\pi(t), \forall t \in T$.

Therefore, we can use the rejection sampling algorithm and model the latent history of the data generative process. By doing this, we can bypass evaluating the intractable normalization term in the likelihood. Let there be r_i rejected proposals preceeding an accepted sample $y_i, i = 1, 2, \ldots, n$ and denote them as $\mathcal{M}_i = \{t_{i,j}\}_{j=1}^{r_i}$, where r_i itself is a random variable. The joint likelihood for $(y_i, \mathcal{M}_i)$ is:

$$p_\pi(y_i, \mathcal{M}_i \mid \boldsymbol{\omega}) = \pi(y_i)\Phi(\omega(y_i)) \left[\prod_{j=1}^{r_i} \pi(t_{i,j}) \left(1 - \Phi\left(\omega(t_{i,j})\right)\right) \right], \tag{2.3}$$

where p_π specifies the choice of the base probability density, which is also the proposal distribution in the rejection sampling algorithm.

Our focus here is to recover the rejected proposals $\mathcal{M}_i$ for each observation $y_i, i = 1, 2, \ldots, n$. Theorem 1 offers us a theoretical foundation to recover the rejected proposals for each observation. Algorithm 1 gives us a convenient way to sample from $p(\mathcal{M}_i \mid x_i)$ (Rao et al., 2016).

Theorem 1. *The set of rejected proposals $\mathcal{M}_i$ preceeding an accepted sample y_i is independent of y_i. And thus assign y_i the set $\hat{\mathcal{M}}$ of another sample $\hat{y}$.*

Algorithm 1: Recovering rejected proposals

Input: A sample y_i, a base probability density $\pi(\cdot)$, a given function $\omega(\cdot)$ for density estimation and a link function $\Phi(\cdot)$

Output: The set of rejected proposals $\mathcal{M}_i$ preceeding y_i

Draw sample $t_{i,j}$ independently from $\pi(t), t \in T$ until an $\hat{y}$ is accepted.

Discard $\hat{y}$ and let $\mathcal{M}_i = \{t_{i,j}\}_{j=1}^{r_i}$.

return $\mathcal{M}_i = \{t_{i,j}\}_{j=1}^{r_i}$

So the latent histories corresponding to the observed data $\mathcal{D}$ includes:

- The values of the latent function $\omega(\cdot)$ at the observed data, denoted $\Omega_n = \{\omega(y_i)\}_{i=1}^n$;
- The number of rejected proposals preceeding each observed data point $r_i, i = 1, 2, \ldots, n$;
- The locations of the r_i rejected preceeding y_i, denoted $\mathcal{M}_i = \{t_{i,j}\}_{j=1}^{r_i}$;
- The values of the latent function $\omega(\cdot)$ at the r_i rejected proposals preceeding y_i, denoted $\Omega_{r_i} = \{\omega(t_{i,j})\}_{j=1}^{r_i}$.

The joint likelihood over both the observed data and the rejected proposals is

$$\begin{aligned} & p\left(\mathcal{D}, \{\mathcal{M}_i\}_{i=1}^{r_i} \mid \boldsymbol{\omega}\right) \\ =& \prod_{i=1}^{n} p_\pi(y_i, \mathcal{M}_i \mid \boldsymbol{\omega}) \\ =& \prod_{i=1}^{n} \left\{ \pi(y_i)\Phi\left(\omega(y_i)\right) \left[\prod_{j=1}^{r_i} \pi(t_{i,j}) \left(1 - \Phi\left(\omega(t_{i,j})\right)\right) \right] \right\}. \end{aligned} \tag{2.4}$$

2.2.2 Pólya-Gamma Representation

With the help of rejection sampling algorithm, we successfully avoid evaluating the normalization term in the likelihood function. However, since the likelihood of the GP variables isn't conjugate to the prior, (Donner & Opper, 2018) has pointed out that a Metropolis-Hastings approach will be time-consuming for this task. Instead, we will use a novel method via representing the logistic function as an infinite mixture of Gaussians involving Pólya-Gamma random variables (Polson et al., 2013) to further augment the model in the way that our model becomes tractable using a simple Gibbs sampler.

As discussed in (Donner & Opper, 2018), the logistic function has a Gaussian represen-

tation as

$$\sigma(z) = \int_0^\infty \exp\left(f(w,z)\right) p_{\mathrm{PG}}(w|1,0)dw, \tag{2.5}$$

where $p_{\mathrm{PG}}(w \mid b,c)$ is the probability density function of the Pólya-Gamma random variable with parameters $b > 0, c \in \mathbb{R}$ and f is a fixed function which is in the form of $f(w,z) = \frac{z}{2} - \frac{z^2}{2}w - \ln(2)$. This conclusion will help us transform the terms $\Phi\left(\omega(y_i)\right) = \sigma\left(\omega(y_i)\right)$ into a Gaussian form. By the property of the logistic function, we have $1 - \Phi\left(\omega(y_i)\right) = 1 - \sigma\left(\omega(y_i)\right) = \sigma\left(-\omega(y_i)\right)$.

Introducing the Pólya-Gamma random variable, we obtain the final augmented joint likelihood for our density estimation task, which is as follows

$$\begin{aligned}
&p\left(\mathcal{D}, \mathcal{W}_n, \{\mathcal{M}_i\}_{i=1}^n, \{\mathcal{W}_{M_i}\}_{i=1}^n \mid \boldsymbol{\omega}\right) \\
=&\prod_{i=1}^n \Bigg\{ \pi(y_i) \exp\left(f(w_i, \omega(y_i))\right) p_{\mathrm{PG}}(w_i \mid 1,0) \\
&\times \left[\prod_{j=1}^{r_i} \pi(t_{i,j}) \exp\left(f(w_{i,j}, -\omega(t_{i,j}))\right) p_{\mathrm{PG}}(w_{i,j} \mid 1,0)\right] \Bigg\},
\end{aligned} \tag{2.6}$$

where $\mathcal{W}_n = \{w_i\}_{i=1}^n$ is the set of latent Pólya-Gamma random variables corresponding to the logistic augmentation at observed data point $y_i, i = 1,2,\ldots,n$. Similarly, we denote the set of latent Pólya-Gamma random variables resulting from the augmentation at the rejected proposals preceeding y_i as $\mathcal{W}_{M_i} = \{w_{i,j}\}_{j=1}^{r_i}$.

It is not hard to verify that the joint likelihood under this data augmentation technique will lead us to the marginal likelihood $p\left(\mathcal{D}, \{\mathcal{M}_i\}_{i=1}^n \mid \boldsymbol{\omega}\right)$ in Equation 2.4. The detailed

proof is as follows

$$
\begin{aligned}
&\int p\left(\mathcal{D}, \mathcal{W}_n, \{\mathcal{M}_i\}_{i=1}^n, \{\mathcal{W}_{M_i}\}_{i=1}^n \mid \boldsymbol{\omega}\right) d\mathcal{W}_n d\mathcal{W}_{M_1} \ldots d\mathcal{W}_{M_n} \\
=&\prod_{i=1}^n \int_{w_i} p_{\mathrm{PG}}(w_i \mid 1,0) e^{(f(w_i,\omega(y_i)))} \pi(y_i) dw_i \prod_{j=1}^{r_i} \int_{w_{i,j}} p_{\mathrm{PG}}(w_{i,j} \mid 1,0) e^{(f(w_{i,j},-\omega(t_{i,j})))} \pi(t_{i,j}) dw_{i,j} \\
=&\prod_{i=1}^n \int_{w_i} p_{\mathrm{PG}}(w_i \mid 1,0) e^{(f(w_i,\omega(y_i)))} \pi(y_i) dw_i \prod_{j=1}^{r_i} \int_{w_{i,j}} p_{\mathrm{PG}}(w_{i,j} | 1,0) e^{(f(w_{i,j},-\omega(t_{i,j})))} \pi(t_{i,j}) dw_{i,j} \\
=&\prod_{i=1}^n \left\{ \pi(y_i) \Phi\left(\omega(y_i)\right) \left[\prod_{j=1}^{r_i} \pi(t_{i,j}) \left(1 - \Phi\left(\omega(t_{i,j})\right)\right) \right] \right\}.
\end{aligned}
\tag{2.7}
$$

2.3 Posterior Inference

In this part, we will give the augmented posterior based on Equation 2.6 as well as the full conditionals for all unknowns to implement an efficient Gibbs sampler.

Based on Equation 2.6, we have the augmented posterior over $\boldsymbol{\omega}$, the rejected proposals $\{\mathcal{M}_i\}_{i=1}^n$ preceeding each observed data, and the Pólya-Gamma variables at the observed data as well as the rejected proposals (which include both $\mathcal{W}_n$ and $\{\mathcal{W}_{M_i}\}_{i=1}^n$) as

$$
\begin{aligned}
&p(\boldsymbol{\omega}, \mathcal{W}_n, \{\mathcal{M}_i\}_{i=1}^n, \{\mathcal{W}_{M_i}\}_{i=1}^n \mid \mathcal{D}) \\
\propto& p\left(\mathcal{D}, \mathcal{W}_n, \{\mathcal{M}_i\}_{i=1}^n, \{\mathcal{W}_{M_i}\}_{i=1}^n \mid \boldsymbol{\omega}\right) p(\boldsymbol{\omega} \mid \theta) \\
=&\prod_{i=1}^n \left\{ \pi(y_i) e^{f(w_i,\omega(y_i))} p_{\mathrm{PG}}(w_i \mid 1,0) \left[\prod_{j=1}^{r_i} \pi(t_{i,j}) \left(e^{f(w_{i,j},-\omega(t_{i,j}))} p_{\mathrm{PG}}(w_{i,j} \mid 1,0) \right) \right] \right\} \\
&\times p(\boldsymbol{\omega} \mid \theta).
\end{aligned}
\tag{2.8}
$$

Based on Equation 2.8, we can proceed to do Bayesian inference via a simple Gibbs sampler.

Gibbs sampling (Geman & Geman, 1984) draws samples from the posterior distribution over all unknowns iteratively. At each iteration, a block of variables is sampled from the full conditional distribution. The detailed derivation for each unknown's full conditional is as follows. Pseudo code for the Gibbs sampler is provided in 2.

2.3.1 Sampling Pólya-Gamma Variables at Observations

Firstly, the full conditional for the Pólya-Gamma variables $\mathcal{W}_n$ at all observations $\mathcal{D} = \{y_i\}_{i=1}^n$ is

$$\begin{aligned}
&p(\mathcal{W}_n \mid \boldsymbol{\omega}, \{\mathcal{M}_i\}_{i=1}^n, \{\mathcal{W}_{M_i}\}_{i=1}^n, \mathcal{D}) \\
\propto &\prod_{i=1}^n [\exp(f(w_i, \omega(y_i))) p_{\mathrm{PG}}(w_i \mid 1, 0)] \\
= &\prod_{i=1}^n \left[\exp\left(\frac{\omega(y_i)}{2} - \frac{\omega(y_i)^2 w_i}{2} - \ln(2)\right) p_{\mathrm{PG}}(w_i \mid 1, 0)\right] \\
= &\prod_{i=1}^n \left[\exp\left(-\frac{\omega(y_i)^2 w_i}{2}\right) p_{\mathrm{PG}}(w_i \mid 1, 0)\right] \\
= &\prod_{i=1}^n p_{\mathrm{PG}}\left(w_i \mid 1, \omega(y_i)\right).
\end{aligned} \tag{2.9}$$

Then the full conditional distribution for each Pólya-Gamma variable at observations is $w_i \sim p_{\mathrm{PG}}(\cdot \mid 1, \omega(y_i))$.

2.3.2 Jointly Sampling Rejected Proposals and Their Pólya-Gamma Variables

As we can see in the augmented posterior, the full conditionals for the rejected proposals $\{\mathcal{M}_i\}_{i=1}^n$ involve their corresponding Pólya-Gamma variables. Likewise, the full conditionals for the Pólya-Gamma variables at the rejected proposals cannot get rid of the rejected proposals. Moreover, the number of the rejected proposals also vary from iteration to iteration, which makes it impossible to draw the rejected proposals and their Pólya-Gamma variables from their full conditionals separately. We need to regard them as a block $\left(\{\mathcal{M}_i\}_{i=1}^n, \{\mathcal{W}_{M_i}\}_{i=1}^n\right)$ in the Gibbs sampling.

Specifically, in order to sample $(\mathcal{M}_k, \mathcal{W}_{M_k})$ from its full conditional distribution, we propose a two-stage strategy as follows:

- Firstly, we draw $t_{k,1}, t_{k,2}, \ldots, t_{k,r_k}$ from $p(\mathcal{M}_k \mid \boldsymbol{\omega}, \mathcal{W}_k, \{\mathcal{M}_i\}_{i \neq k}, \mathcal{D})$;
- Secondly, based on the rejected proposals, we can further draw the corresponding

Pólya-Gamma variables from the full conditionals, which we will give detailed derivation below.

Specifically, drawing $t_{k,1}, t_{k,2}, \ldots, y_{k,r_k}$ from $p(\mathcal{M}_k \mid \boldsymbol{\omega}, \mathcal{W}_n, \{\mathcal{M}_i\}_{i\neq k}, \mathcal{D})$ can be easily achieved through Algorithm 1 which we mentioned above. Since

$$\begin{aligned}
&p(\mathcal{M}_k \mid \boldsymbol{\omega}, \mathcal{W}_n, \{\mathcal{M}_i\}_{i\neq k}, \mathcal{D}) \\
=&\int p(\mathcal{M}_k, \mathcal{W}_{M_k} \mid \boldsymbol{\omega}, \mathcal{W}_n, \{\mathcal{M}_i\}_{i\neq k}, \mathcal{D}) d\mathcal{W}_{M_k} \\
=&\prod_{j=1}^{r_k} \left[\int \pi(t_{k,j}) \exp(f(w_{k,j}, -\omega(t_{k,j}))) p_{\mathrm{PG}}(w_{k,j} \mid 1, 0) dw_{k,j} \right] \\
=&\prod_{j=1}^{r_k} \left[\pi(t_{k,j})(1 - \Phi(\omega(t_{k,j}))) \right].
\end{aligned} \tag{2.10}$$

Having obtained this probability density over any potential sequences of rejected proposals preceeding y_k, we can observe that this distribution just has the same kernel as $p(\mathcal{M}_k \mid y_k)$. And that's why it's reasonable to draw the rejected proposals via Algorithm 1.

Similar to the last part, the full conditional for the Pólya-Gamma variables $\mathcal{W}_{M_i}$ at rejected proposals preceeding observation $y_i, i = 1, 2, \ldots, n$ is

$$\begin{aligned}
&p(\mathcal{W}_{M_k} \mid \boldsymbol{\omega}, \mathcal{W}_n, \{\mathcal{M}_i\}_{i=1}^n, \{\mathcal{W}_{M_i}\}_{i\neq k}, \mathcal{D}) \\
\propto&\prod_{j=1}^{r_k} \left[\exp(f(w_{k,j}, -\omega(t_{k,j}))) p_{\mathrm{PG}}(w_{k,j} \mid 1, 0) \right] \\
=&\prod_{j=1}^{r_k} \left[\exp\left(-\frac{\omega(t_{k,j})}{2} - \frac{\omega(t_{k,j})^2 w_{k,j}}{2} - \ln(2) \right) p_{\mathrm{PG}}(w_{k,j} \mid 1, 0) \right] \\
=&\prod_{j=1}^{r_k} \left[\exp\left(-\frac{\omega(t_{k,j})^2 w_{k,j}}{2} \right) p_{\mathrm{PG}}(w_{k,j} \mid 1, 0) \right] \\
=&\prod_{j=1}^{r_k} p_{\mathrm{PG}}\left(w_{k,j} \mid 1, \omega(t_{k,j})\right).
\end{aligned} \tag{2.11}$$

Therefore, the full conditional distribution for a single Pólya-Gamma variable at a specific

rejected proposal $t_{i,j}$ is $w_{i,j} \sim p_{\mathrm{PG}}(\cdot \mid 1, \omega(t_{i,j}))$.

2.3.3 Sampling Gaussian Processes

In this part, we need to sample the values of $\omega(t)$ at both observations $\mathcal{D} = \{y_i\}_{i=1}^n$ and rejected proposals $\{\mathcal{M}_i\}_{i=1}^n$ from their full conditionals. As we've discussed, introducing Pólya-Gamma variables for each data point (both observations and rejected proposals) allows directly generating data points from the normal distribution. So all we need to do is to derive the mean and covariance for the full conditional over $\omega(t)$'s values at these data

points, which is a multivariate Gaussian density as follows:

$$
\begin{aligned}
& p\left(\Omega_n, \{\Omega_{M_i}\}_{i=1}^n \mid \mathcal{W}_n, \{\mathcal{M}_i\}_{i=1}^n, \{\mathcal{W}_i\}_{i=1}^n, \mathcal{D}\right) \\
\propto & \prod_{i=1}^n \left\{ \exp(f(w_i, \omega(y_i))) \left[\prod_{j=1}^{r_i} \exp(f(w_{i,j}, -\omega(t_{i,j}))) \right] \right\} \\
& \times \mathcal{GP}(\Omega_n, \{\Omega_{M_i}\}_{i=1}^n \mid \mathcal{D}, \{\mathcal{M}_i\}_{i=1}^n; \theta) \\
\propto & \prod_{i=1}^n \left\{ \exp\left(\frac{\omega(y_i)}{2} - \frac{\omega(y_i)^2 w_i}{2} \right) \left[\prod_{j=1}^{r_i} \exp\left(-\frac{\omega(t_{i,j})}{2} - \frac{\omega(t_{i,j})^2 w_{i,j}}{2} \right) \right] \right\} \\
& \times \mathcal{GP}(\Omega_n, \{\Omega_{M_i}\}_{i=1}^n \mid \mathcal{D}, \{\mathcal{M}_i\}_{i=1}^n; \theta) \\
\propto & \exp\left[\sum_{i=1}^n \left(\frac{\omega(y_i)}{2} - \frac{\omega(y_i)^2 w_i}{2} \right) + \sum_{i=1}^n \sum_{j=1}^{r_i} \left(\frac{-\omega(t_{i,j})}{2} - \frac{\omega(t_{i,j})^2 w_{i,j}}{2} \right) \right] \\
& \times \mathcal{GP}(\Omega_n, \{\Omega_{M_i}\}_{i=1}^n \mid \mathcal{D}, \{\mathcal{M}_i\}_{i=1}^n; \theta) \\
= & \exp\left(-\frac{1}{2} \boldsymbol{\omega}_{\text{obs-rej}}^T \Lambda \boldsymbol{\omega}_{\text{obs-rej}} + \boldsymbol{\omega}_{\text{obs-rej}}^T \mathbf{c} \right) \\
& \times \mathcal{GP}(\Omega_n, \{\Omega_{M_i}\}_{i=1}^n \mid \mathcal{D}, \{\mathcal{M}_i\}_{i=1}^n; \theta) \\
= & \exp\left(-\frac{1}{2} \boldsymbol{\omega}_{\text{obs-rej}}^T \Lambda \boldsymbol{\omega}_{\text{obs-rej}} + \boldsymbol{\omega}_{\text{obs-rej}}^T \mathbf{c} \right) \\
& \times \exp\left(-\frac{1}{2} \left(\boldsymbol{\omega}_{\text{obs-rej}} - \mu_0 \mathbf{1}_{\text{obs-rej}} \right)^T K_{\text{obs-rej}}^{-1} \left(\boldsymbol{\omega}_{\text{obs-rej}} - \mu_0 \mathbf{1}_{\text{obs-rej}} \right) \right) \\
\propto & \exp\left(-\frac{1}{2} (\boldsymbol{\omega}_{\text{obs-rej}} - \mu)^T \Sigma^{-1} (\boldsymbol{\omega}_{\text{obs-rej}} - \mu) \right),
\end{aligned}
\tag{2.12}
$$

where

$$
\boldsymbol{\omega}_{\text{obs-rej}} = \left(\omega(y_1), \ldots, \omega(y_n), \omega(t_{1,1}), \ldots, \omega(t_{1,r_1}), \ldots, \omega(t_{n,1}), \ldots, \omega(t_{n,r_n}) \right)^T
$$

is the vector of values of $\omega(\cdot)$ sampled at both observations $\mathcal{D}$ and rejections $\{\mathcal{M}_i\}_{i=1}^n$. Let $M = \sum_{i=1}^n r_i$, Λ is a diagonal matrix with the beginning n entries to be the Pólya-Gamma variables $\mathcal{W}_n$ at observations, and the last M entries to be the Pólya-Gamma variables $\{\mathcal{W}_i\}_{i=1}^n$ at rejections $\{\mathcal{M}_i\}_{i=1}^n$. For our application, we assume μ_0 to be the prior mean of

Gaussian process over the whole space $\mathcal{X}$, which is a constant. $\mathbf{1}_{\text{obs-rej}}$ denotes the $n+M$ vector with all entries to be 1. $\mathbf{c} = (\underbrace{\frac{1}{2},\ldots,\frac{1}{2}}_{n},\underbrace{-\frac{1}{2},\ldots,-\frac{1}{2}}_{r_1},\ldots,\underbrace{-\frac{1}{2},\ldots,-\frac{1}{2}}_{r_n})^T$ denotes a constant $(n+M)-$ dimensional vector, with the begining n entries to be $\frac{1}{2}$, and the last M entries to be $-\frac{1}{2}$. $K_{\text{obs-rej}}$ is the prior covariance matrix computed at both observations $\mathcal{D}$ and rejections $\{\mathcal{M}_i\}_{i=1}^n$. Moreover, $\Sigma = (K_{\text{obs-rej}}^{-1} + \Lambda)^{-1}$ and $\mu = \Sigma(K_{\text{obs-rej}}^{-1}(\mu_0 \mathbf{1}_{\text{obs-rej}}) + \mathbf{c})$.

Algorithm 2: Gibbs sampler for Bayesian density model using rejection sampling

Input: The number of MCMC iterations S, observations $\mathcal{D} = \{y_i\}_{i=1}^n$, the Gaussian process covariance function $k(u, v; \theta)$, a base probability density $\pi(\cdot)$

Output: The set of rejected proposals $\{\mathcal{M}_i\}_{i=1}^n$ preceeding observations $\mathcal{D}$, the set of Pólya-Gamma variables $\{\mathcal{W}_{M_i}\}_{i=1}^n$ corresponding to rejected proposals $\{\mathcal{M}_i\}_{i=1}^n$, Pólya-Gamma variables $\mathcal{W}_n$ corresponding to observations $\mathcal{D}$, the values of posterior GP at observations Ω_n and the values of posterior GP at all rejected proposals $\{\Omega_{M_i}\}_{i=1}^n$

for $t \leftarrow 1$ *to* S **do**
- Sample observations $\mathcal{D}$'s corresponding Pólya-Gamma variables: $\mathcal{W}_n \sim$ Equation 2.9
- Sample GP at both observations $\mathcal{D}$ and rejected proposals $\{\mathcal{M}_i\}_{i=1}^n$: $\left(\Omega_n, \{\Omega_{M_i}\}_{i=1}^n\right) \sim$ Equation 2.12
- Jointly sample the rejected proposals as well as their corresponding Pólya-Gamma variables $(\mathcal{M}_k, \mathcal{W}_{M_k})$ for each observation via our proposed two-stage strategy as follows: **for** $k \leftarrow 1$ *to* n **do**
  - Sample the rejected proposals $\mathcal{M}_k$ preceeding observation y_k via Algorithm 1
  - Sample $\mathcal{M}_k$'s corresponding Pólya-Gamma variables: $\mathcal{W}_{M_k} \sim$ Equation 2.11

return the samples for $\{\mathcal{M}_i\}_{i=1}^n$, $\{\mathcal{W}_{M_i}\}_{i=1}^n$, $\mathcal{W}_n$, Ω_n and $\{\Omega_{M_i}\}_{i=1}^n$

When doing experiments on the simulated datasets, we find that the Gibbs sampler via rejection sampling could be much more efficient if we could make full use of the computation advantage of vectorization. Therefore, we hope to implement the process of rejection sampling for a batch of samples at a time rather than just one at a time, which could also take advantage of the benefit of vectorization. However, since we have no idea how many data points we need to sample from the proposal distribution in order to obtain adequate accepted proposals. One strategy is to start with the batch size which equals the number of observations we have. Then we sample the GP values of this batch and decide which should

be accepted or rejected. As we've described in Algorithm 1, the set of rejected proposals $\mathcal{M}_i$ preceeding an accepted sample y_i is independent of y_i. And thus assign y_i the set $\hat{\mathcal{M}}$ of another sample $\hat{y}$. Therefore, we don't need to pay attention to the correspondence between the rejected proposals and observations. All we need to do is to check if we have collect enough accepted proposals during this process. As long as we've, only rejected proposals' locations as well as the corresponding GP values in this iteration will be kept.

3. Tree-logistic-link Gaussian Process Model

This chapter gives technical details and some properties of the tree-logistic-link Gaussian process (TLLGP).

3.1 Tree-logistic-link Gaussian Process Model

Suppose $\pi(t)$ is a probability density function and $\omega(t)$ is any continuous function on a fixed bounded interval $T \subseteq \mathbb{R}^d$ with $d \geqslant 1$. Without loss of generality, we could take $T = [0,1]^d$. Consider a random draw y from T returned by the following algorithm:

- make two independent draws $t_1, t_2 \in T$ according to the density $\pi(t)$;
- return $y = t_1$ with probability $p_1 = \frac{e^{\omega(t_1)}}{e^{\omega(t_1)}+e^{\omega(t_2)}}$, otherwise return $y = t_2$.

As a result, a random variable generated as above has probability density function

$$\begin{aligned}\tilde{\pi}(t) &= 2\int_T \frac{e^{\omega(t)}}{e^{\omega(t)}+e^{\omega(s)}}\pi(t)\pi(s)ds \\ &= 2\pi(t)\int_T \frac{1}{1+e^{\omega(s)-\omega(t)}}\pi(s)ds, \quad t \in T.\end{aligned} \tag{3.1}$$

We denote the map from the probability density function π to another probability density function $\tilde{\pi}$ by L_ω, and define a depth-m tree-logistic-link Gaussian process (TLLGP) η as $\eta = L_\omega^m \pi_0$, where π_0 is the uniform distribution on T with continuous sample paths. Consider a fixed mean function $\mu(\cdot)$ on T and a family of covariance functions $\sigma_\lambda(u,v) = \sigma_0(\lambda u, \lambda v)$ on $T \times T$, where λ is a finite dimensional parameter and λu denotes the vector of coordinate-wise products of λ and u when $d > 1$. Take a probability distribution π_λ on the space of λ. Let $\mathcal{GP}(\omega \mid 0, \sigma_\lambda)$ denote the distribution of a separable mean zero Gaussian process on T with covariance $\sigma_\lambda(\cdot,\cdot)$. Then, when given a pre-specified integer m, we can model a random density function f on T as follows:

$$\begin{aligned} f \mid \omega, \lambda &= L_\omega^m \pi_0, \\ \omega \mid \lambda &\sim \mathcal{GP}(\omega \mid 0, \sigma_\lambda), \\ \lambda &\sim \pi_\lambda, \end{aligned} \tag{3.2}$$

which induces a prior on the space of probability densities on T. And we shall call it a tree-logistic-link Gaussian process (TLLGP) prior.

For a better explanation of our approach, below we give a toy example. When $m = 4$ and ω is given, a draw y from the density function $\eta = L_\omega^4 \pi_0$ can be generated as Figure 3.1 indicates. Specifically, we generate each node of the tree from the bottom to the top. For a contest between two points $t_1, t_2 \in T$, we will choose t_1, t_2 with the probability of $\frac{e^{\omega(t_1)}}{e^{\omega(t_1)}+e^{\omega(t_2)}}$ and $\frac{e^{\omega(t_2)}}{e^{\omega(t_1)}+e^{\omega(t_2)}}$ respectively. For illustration, we define $t_1^4, t_3^4, \ldots, t_{15}^4$ representing the proposals winning the contests at the layer-1, and $t_2^4, t_4^4, \ldots, t_{16}^4$ representing the proposals losing the contests at the layer-1. In addition, $t_1^4, t_5^4, t_9^4, t_{13}^4$ are also the proposals winning the contests at the layer-2. As we can observed from Figure 3.1, t_1^4 wins all contests; t_9^4 wins the contests at the first three layers, but loses the contest against t_1^4. In general, with a given ω, a draw y from the corresponding density function $\eta = L_\omega^m \pi_0$ can be obtained by carrying out Algorithm 3 which expands the above two-point sampling strategy onto a binary tree of depth m.

Algorithm 3: Implementing a depth-m binary tree

Input: Tree depth m, a function ω

Output: A random draw y from the density $\eta = L_\omega^m \pi_0$, the correspond set of proposals $\mathcal{T}^m = \left\{t_1^m, t_2^m, \ldots, t_{2^m}^m\right\}$

Let $\mathcal{T}^m$ be an empty set.
Draw 2^m points $t_{0,1}, t_{0,2}, \ldots, t_{0,2^m}$ uniformly from T
Set $(s_{0,k}, r_{0,k}) \leftarrow (\exp[\omega(t_{0,k})], \texttt{NULL})$, $k = 1, 2, \ldots, 2^m$
for $l \leftarrow 1$ *to* m **do**
    for $k \leftarrow 1$ *to* 2^{m-l} **do**
        Calculate $p_1 \leftarrow \frac{s_{l-1,2k-1}}{s_{l-1,2k-1}+s_{l-1,2k}}$
        Draw $u \sim \text{Uniform}(0,1)$
        if $u < p_1$ **then**
            Set $(t_{l,k}, s_{l,k}) \leftarrow (t_{l-1,2k-1}, s_{l-1,2k-1})$
            Set $(r_{l-1,2k-1}, r_{l-1,2k}) \leftarrow (\texttt{TRUE}, \texttt{FALSE})$
        else
            Set $(t_{l,k}, s_{l,k}) \leftarrow (t_{l-1,2k}, s_{l-1,2k})$
            Set $(r_{l-1,2k-1}, r_{l-1,2k}) \leftarrow (\texttt{FALSE}, \texttt{TRUE})$
for $l \leftarrow m$ *to* 1 **do**
    for $k \leftarrow 1$ *to* 2^{m-l} **do**
        while $r_{l-1,2k}$ **do**
            for $l' \leftarrow 0$ *to* $l-1$ **do**
                for $k' \leftarrow 1$ *to* $2^{l-1-l'}$ **do**
                    Swap $\left(t_{l',k'}, s_{l',k'}, r_{l',k'}\right)$ and $\left(t_{l',k'+2^{l-1-l'}}, s_{l',k'+2^{l-1-l'}}, r_{l',k'+2^{l-1-l'}}\right)$
Set $t_k^m \leftarrow t_{0,k}$, $k = 1, 2, \ldots, 2^m$
return $y = t_{0,1}$ and $\mathcal{T}^m = \{t_1^m, \ldots, t_{2^m}^m\}$

3.2 Posterior Computation

3.2.1 Data Augmentation

For simplicity, we start with the case of only one observation $y \in T$. As we've mentioned above, if we assume observations are generated from $\eta = L_\omega^m \pi_0$, $m > 0$, there will be $2^m - 1$ rejected proposals as well as m contests associated with y. For illustration, we denote competitors in m contests by $t^0, t^1, \ldots, t^{m-1}$. As a result, we can easily write out the joint likelihood function over y and $t^0, t^1, \ldots, t^{m-1}$ and the augmented posterior distribution over ω, λ and $t^0, t^1, \ldots, t^{m-1}$ given observation y as follow,

$$p\left(t^0, t^1, \ldots, t^{m-1}, \omega, \lambda \mid y\right) \propto p\left(y, t^0, t^1, \ldots, t^{m-1} \mid \omega\right) \times \mathcal{GP}(\omega \mid 0, \sigma_\lambda)\pi_\lambda, \tag{3.3}$$

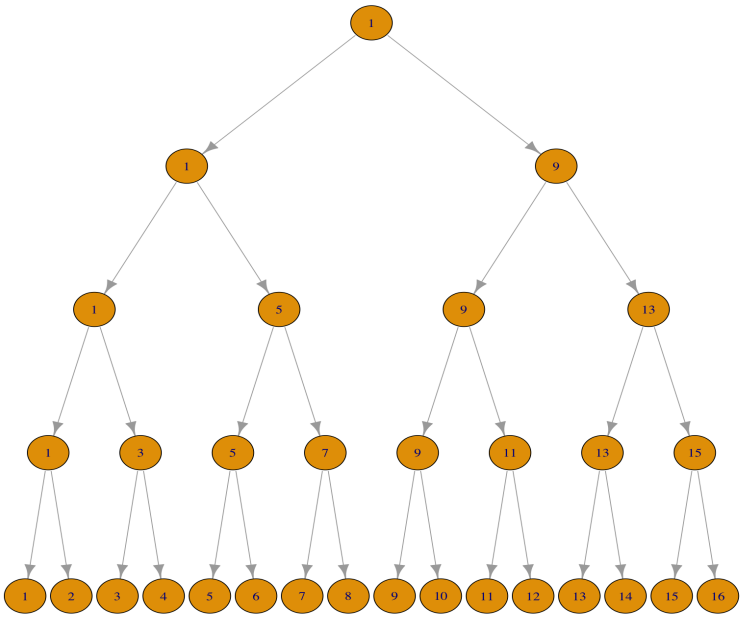

FIGURE 3.1: A depth-4 binary tree

where

$$p\left(y, t^0, t^1, \ldots, t^{m-1} \mid \omega\right) = \prod_{\tilde{m}=0}^{m-1} \left\{ L_\omega^{\tilde{m}} \pi_0\left(t^{\tilde{m}}\right) \times \frac{e^{\omega(y)}}{e^{\omega(y)} + e^{\omega(t^{\tilde{m}})}} \right\} \tag{3.4}$$

As implied by Equation (3.3), the full conditional distribution of ω involves the terms $L_\omega^{\tilde{m}}$, $\tilde{m} = 0, 1, \ldots, m-1$. Therefore, we have to expand each $L_\omega^{\tilde{m}}$ to sample ω from its full conditional.

3.2.1.1 Expanding a Depth-m Binary Tree

By the notations defined in Algorithm 3, for a depth-$\tilde{m}$ binary tree $\mathcal{T}^{\tilde{m}} = \{t_1^{\tilde{m}}, t_2^{\tilde{m}}, \ldots, t_{2^{\tilde{m}}}^{\tilde{m}}\}$, when $\tilde{m} > 0$ and ω is given, the data likelihood function over $t_1^{\tilde{m}}, t_2^{\tilde{m}}, \ldots, t_{2^{\tilde{m}}}^{\tilde{m}}$ is

$$p\left(t_1^{\tilde{m}}, t_2^{\tilde{m}}, \ldots, t_{2^{\tilde{m}}}^{\tilde{m}} \mid \omega\right) = \prod_{l=1}^{\tilde{m}} \left\{ \prod_{k=1}^{2^{\tilde{m}-l}} \frac{\exp\left(\omega\left(t_{1+(k-1)2^l}^{\tilde{m}}\right)\right)}{\exp\left(\omega\left(t_{1+(k-1)2^l}^{\tilde{m}}\right)\right) + \exp\left(\omega\left(t_{1+(k-1)2^l+2^{l-1}}^{\tilde{m}}\right)\right)} \right\}. \tag{3.5}$$

After marginalizing out other rejected proposals $t_2^{\tilde{m}}, t_3^{\tilde{m}}, \ldots, t_{2^{\tilde{m}}}^{\tilde{m}}$, we obtain the marginal distribution $L_\omega^{\tilde{m}}$ as follows

$$\begin{aligned} L_\omega^{\tilde{m}}\left(t_1^{\tilde{m}}\right) &= \int_{t_2^{\tilde{m}}} \cdots \int_{t_{2^{\tilde{m}}}^{\tilde{m}}} p\left(t_1^{\tilde{m}}, t_2^{\tilde{m}}, \ldots, t_{2^{\tilde{m}}}^{\tilde{m}} \mid \omega\right) dt_2^{\tilde{m}} \ldots dt_{2^{\tilde{m}}}^{\tilde{m}} \\ &= \int_{t_2^{\tilde{m}}} \cdots \int_{t_{2^{\tilde{m}}}^{\tilde{m}}} \prod_{l=1}^{\tilde{m}} \left\{ \prod_{k=1}^{2^{\tilde{m}-l}} \frac{e^{\omega\left(t_{1+(k-1)2^l}^{\tilde{m}}\right)}}{e^{\omega\left(t_{1+(k-1)2^l}^{\tilde{m}}\right)} + e^{\omega\left(t_{1+(k-1)2^l+2^{l-1}}^{\tilde{m}}\right)}} \right\} dt_2^{\tilde{m}} \ldots dt_{2^{\tilde{m}}}^{\tilde{m}}. \end{aligned} \tag{3.6}$$

However, it's evident that this marginal distribution is hard to deal with in Bayesian inference. Instead, we tend to write out the joint likelihood over all the rejected proposals. As a result, the augmented likelihood should be

$$\begin{aligned} & p\left(y, \mathcal{T}^0, \mathcal{T}^1, \ldots, \mathcal{T}^{m-1} \mid \omega\right) \\ &= \prod_{\tilde{m}=0}^{m-1} \left\{ p\left(t_1^{\tilde{m}}, t_2^{\tilde{m}}, \ldots, t_{2^{\tilde{m}}}^{\tilde{m}} \mid \omega\right) \times \frac{e^{\omega(y)}}{e^{\omega(y)} + e^{\omega\left(t_1^{\tilde{m}}\right)}} \right\} \\ &= \prod_{\tilde{m}=1}^{m-1} \left\{ \prod_{l=1}^{\tilde{m}} \left\{ \prod_{k=1}^{2^{\tilde{m}-l}} \frac{e^{\omega\left(t_{1+(k-1)2^l}^{\tilde{m}}\right)}}{e^{\omega\left(t_{1+(k-1)2^l}^{\tilde{m}}\right)} + e^{\omega\left(t_{1+(k-1)2^l+2^{l-1}}^{\tilde{m}}\right)}} \right\} \frac{e^{\omega(y)}}{e^{\omega(y)} + e^{\omega\left(t_1^{\tilde{m}}\right)}} \right\} \times \frac{e^{\omega(y)}}{e^{\omega(y)} + e^{\omega\left(t_1^0\right)}} \\ &= \prod_{\tilde{m}=1}^{m-1} \left\{ \prod_{l=1}^{\tilde{m}} \prod_{k=1}^{2^{\tilde{m}-l}} \Phi\left\{\omega\left(t_{1+(k-1)2^l}^{\tilde{m}}\right) - \omega\left(t_{1+(k-1)2^l+2^{l-1}}^{\tilde{m}}\right)\right\} \Phi\left\{\omega(y) - \omega\left(t_1^{\tilde{m}}\right)\right\} \right\} \\ &\quad \times \Phi\left\{\omega(y) - \omega\left(t_1^0\right)\right\}, \end{aligned} \tag{3.7}$$

where $\Phi(z)$ is the sigmoid function.

3.2.1.2 Pólya-Gamma Augmentation

As implied by Equation (3.7), the augmented likelihood of ω isn't conjugate to the prior, which again motivates us to introduce Pólya-gamma variables (Polson et al., 2013), and further augment the model in a way that our model becomes tractable with a simple Gibbs sampler. (Donner & Opper, 2018) proposes to represent the logistic function as $\Phi(z) = \int_0^\infty \mathrm{e}^{h(w,z)} p_{\mathrm{PG}}(w|1,0)dw$, where $p_{\mathrm{PG}}(w|b,c)$ is the probability density function of the

Pólya-gamma random variable with parameters $b > 0, c \in \mathbb{R}$ and $h(w,z) = \frac{z}{2} - \frac{z^2}{2}w - \ln 2$. This conclusion helps us transform the sigmoid term into a Gaussian form. After introducing the Pólya-gamma variables, we obtain the final augmented joint likelihood as follows,

$$
\begin{aligned}
& p\left(y, \mathcal{T}^0, \mathcal{W}^0, \mathcal{T}^1, \mathcal{W}^1, \ldots, \mathcal{T}^{m-1}, \mathcal{W}^{m-1} \mid \omega\right) \\
= & \prod_{\tilde{m}=1}^{m-1} \left\{ \prod_{l=1}^{\tilde{m}} \left(\prod_{k=1}^{2^{\tilde{m}-l}} e^{h\left(w^{\tilde{m}}_{\left(1+(k-1)2^l, 1+(k-1)2^l+2^{l-1}\right)}, \omega\left(t^{\tilde{m}}_{1+(k-1)2^l}\right) - \omega\left(t^{\tilde{m}}_{1+(k-1)2^l+2^{l-1}}\right)\right)} \right.\right. \\
& \left.\left. \times\, p_{\mathrm{PG}}\left(w^{\tilde{m}}_{(1+(k-1)2^l, 1+(k-1)2^l+2^{l-1})} \mid 1, 0\right) \right) \times e^{h\left(w^{\tilde{m}}, \omega(y) - \omega\left(t_1^{\tilde{m}}\right)\right)} p_{\mathrm{PG}}\left(w^{\tilde{m}} \mid 1, 0\right) \right\} \\
& \times e^{h\left(w^0, \omega(y) - \omega\left(t_1^0\right)\right)} p_{\mathrm{PG}}\left(w^0 \mid 1, 0\right)
\end{aligned}
\tag{3.8}
$$

where $\mathcal{W}^{\tilde{m}} = \left\{ \left\{ w^{\tilde{m}}_{(1+(k-1)2^l, 1+(k-1)2^l+2^{l-1})} \right\}_{k=1}^{2^{\tilde{m}-l}} \right\}_{l=1}^{\tilde{m}} \bigcup \left\{ w^{\tilde{m}} \right\}$, $w^{\tilde{m}}_{(1+(k-1)2^l, 1+(k-1)2^l+2^{l-1})}$ is the Pólya-gamma variable associated with the contest between $t^{\tilde{m}}_{1+(k-1)2^l}$ and $t^{\tilde{m}}_{1+(k-1)2^l+2^{l-1}}$ and $w^{\tilde{m}}$ is the Pólya-gamma variable associated with the contest between observation y and the rejected proposal $t_1^{\tilde{m}}$. Based on Equation (3.8), when there are n observations, we can easily write out the joint likelihood function as follows,

$$
\begin{aligned}
& p(y_1, \mathcal{T}_1, \mathcal{W}_1, y_2, \mathcal{T}_2, \mathcal{W}_2, \ldots, y_n, \mathcal{T}_n, \mathcal{W}_n \mid \omega) \\
= & \prod_{i=1}^{n} \left\{ \prod_{\tilde{m}=1}^{m-1} \prod_{l=1}^{\tilde{m}} \prod_{k=1}^{2^{\tilde{m}-l}} e^{h\left(w^{\tilde{m}}_{i,\left(1+(k-1)2^l, 1+(k-1)2^l+2^{l-1}\right)}, \omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l}\right) - \omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l+2^{l-1}}\right)\right)} \right. \\
& \times p_{\mathrm{PG}}\left(w^{\tilde{m}}_{i,\left(1+(k-1)2^l, 1+(k-1)2^l+2^{l-1}\right)} \mid 1, 0\right) \exp\left\{h\left(w_i^{\tilde{m}}, \omega(y_i) - \omega\left(t_{i,1}^{\tilde{m}}\right)\right)\right\} p_{\mathrm{PG}}\left(w_i^{\tilde{m}} \mid 1, 0\right) \\
& \left. \times \exp\left\{h\left(w_i^0, \omega(y_i) - \omega\left(t_{i,1}^0\right)\right)\right\} p_{\mathrm{PG}}\left(w_i^0 \mid 1, 0\right) \right\},
\end{aligned}
\tag{3.9}
$$

where $\mathcal{T}_i = \left\{\mathcal{T}_i^{\tilde{m}}\right\}_{\tilde{m}=0}^{m-1}$, $\mathcal{T}_i^{\tilde{m}}$ is the depth-$\tilde{m}$ binary tree associated with observation y_i, $\mathcal{W}_i = \left\{\mathcal{W}_i^{\tilde{m}}\right\}_{\tilde{m}=0}^{m-1}$, and $\mathcal{W}_i^{\tilde{m}} = \left\{ \left\{ w^{\tilde{m}}_{i,(1+(k-1)2^l, 1+(k-1)2^l+2^{l-1})} \right\}_{k=1}^{2^{\tilde{m}-l}} \right\}_{l=1}^{\tilde{m}} \bigcup \left\{ w_i^{\tilde{m}} \right\}$.

3.2.2 Gibbs Sampling Algorithm

Based on the final augmented likelihood in Equation (3.9), the joint posterior distribution is

$$
\begin{aligned}
&p\left(\mathcal{T}_1, \mathcal{W}_1, \mathcal{T}_2, \mathcal{W}_2, \ldots, \mathcal{T}_n, \mathcal{W}_n, \omega, \lambda \mid y_1, y_2, \ldots, y_n\right) \\
\propto & p(y_1, \mathcal{T}_1, \mathcal{W}_1, y_2, \mathcal{T}_2, \mathcal{W}_2, \ldots, y_n, \mathcal{T}_n, \mathcal{W}_n \mid \omega) \times \mathcal{GP}(\omega \mid 0, \sigma_\lambda)\pi_\lambda \\
= & \prod_{i=1}^{n} \left\{ \prod_{\tilde{m}=1}^{m-1} \prod_{l=1}^{\tilde{m}} \prod_{k=1}^{2^{\tilde{m}-l}} e^{h\left(w^{\tilde{m}}_{i,\left(1+(k-1)2^l, 1+(k-1)2^l+2^{l-1}\right)}, \omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l}\right) - \omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l+2^{l-1}}\right)\right)} \right. \\
& \times p_{\mathrm{PG}}\left(w^{\tilde{m}}_{i,\left(1+(k-1)2^l, 1+(k-1)2^l+2^{l-1}\right)} \mid 1, 0\right) \exp\left\{h\left(w_i^{\tilde{m}}, \omega(y_i) - \omega\left(t^{\tilde{m}}_{i,1}\right)\right)\right\} p_{\mathrm{PG}}\left(w_i^{\tilde{m}} \mid 1, 0\right) \\
& \left. \times \exp\left\{h\left(w_i^0, \omega(y_i) - \omega\left(t^0_{i,1}\right)\right)\right\} p_{\mathrm{PG}}\left(w_i^0 \mid 1, 0\right) \right\} \times \mathcal{GP}(\omega \mid 0, \sigma_\lambda)\pi_\lambda
\end{aligned}
\tag{3.10}
$$

3.2.2.1 Jointly Sampling Rejected Proposals and Pólya-gamma Variables

For observation y_i, the full conditional of its rejected proposals $\mathcal{T}_i = \left\{\mathcal{T}_i^{\tilde{m}}\right\}_{\tilde{m}=0}^{m-1}$ as well as Pólya-gamma variables $\mathcal{W}_i = \left\{\mathcal{W}_i^{\tilde{m}}\right\}_{\tilde{m}=0}^{m-1}$ is as follows,

$$
\begin{aligned}
& p\left(\mathcal{T}_i, \mathcal{W}_i \mid \left\{\mathcal{T}_{i'}\right\}_{i' \neq i}, \left\{\mathcal{W}_{i'}\right\}_{i' \neq i}, \omega, \lambda, y_1, y_2, \ldots, y_n\right) \\
\propto & \prod_{\tilde{m}=1}^{m-1} \left[\prod_{l=1}^{\tilde{m}} \left(\prod_{k=1}^{2^{\tilde{m}-l}} e^{h\left(w^{\tilde{m}}_{i,\left(1+(k-1)2^l, 1+(k-1)2^l+2^{l-1}\right)}, \omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l}\right) - \omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l+2^{l-1}}\right)\right)} \right. \right. \\
& \left. \left. \times p_{\mathrm{PG}}\left(w^{\tilde{m}}_{i,\left(1+(k-1)2^l, 1+(k-1)2^l+2^{l-1}\right)} \mid 1, 0\right) \right) e^{h\left(w_i^{\tilde{m}}, \omega(y_i) - \omega\left(t^{\tilde{m}}_{i,1}\right)\right)} p_{\mathrm{PG}}\left(w_i^{\tilde{m}} \mid 1, 0\right) \right] \\
& \times \exp\left\{h\left(w_i^0, \omega(y_i) - \omega\left(t^0_{i,1}\right)\right)\right\} p_{\mathrm{PG}}\left(w_i^0 \mid 1, 0\right).
\end{aligned}
\tag{3.11}
$$

In order to jointly sample $(\mathcal{T}_i, \mathcal{W}_i)$ from the full conditional, we could

- firstly sample $\mathcal{T}_i = \left\{\mathcal{T}_i^{\tilde{m}}\right\}_{\tilde{m}=0}^{m-1}$ from $p\left(\mathcal{T}_i \mid \left\{\mathcal{T}_{i'}\right\}_{i' \neq i}, \left\{\mathcal{W}_{i'}\right\}_{i' \neq i}, \omega, \lambda, y_1, \ldots, y_n\right)$,
- and then sample $\mathcal{W}_i = \left\{\mathcal{W}_i^{\tilde{m}}\right\}_{\tilde{m}=0}^{m-1}$ from $p\left(\mathcal{W}_i \mid \left\{\mathcal{T}_{i'}\right\}_{i'=1}, \left\{\mathcal{W}_{i'}\right\}_{i' \neq i}, \omega, \lambda, y_1, \ldots, y_n\right)$.

▷ **Sampling rejected proposals**

Marginalizing out Pólya-gamma variables, we obtain

$$\begin{aligned}
&p\left(\mathcal{T}_i \mid \{\mathcal{T}_{i'}\}_{i'\neq i}, \{\mathcal{W}_{i'}\}_{i'\neq i}, \omega, \lambda, y_1, \ldots, y_n\right) \\
&= \int_{\mathcal{W}_i} p\left(\mathcal{T}_i, \mathcal{W}_i \mid \{\mathcal{T}_{i'}\}_{i'\neq i}, \{\mathcal{W}_{i'}\}_{i'\neq i}, \omega, \lambda, y_1, \ldots, y_n\right) d\mathcal{W}_i \\
&\propto \prod_{\tilde{m}=1}^{m-1} \left[\prod_{l=1}^{\tilde{m}} \prod_{k=1}^{2^{\tilde{m}-l}} \Phi\left\{\omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l}\right) - \omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l+2^{l-1}}\right)\right\} \Phi\left\{\omega(y_i) - \omega\left(t^{\tilde{m}}_{i,1}\right)\right\} \right] \\
&\quad \times \Phi\left\{\omega(y_i) - \omega\left(t^{0}_{i,1}\right)\right\},
\end{aligned} \tag{3.12}$$

which implies that

$$\begin{aligned}
&p\left(\mathcal{T}_i^{\tilde{m}} \mid \{\mathcal{T}_{i'}\}_{i'\neq i}, \{\mathcal{W}_{i'}\}_{i'\neq n}, \omega, \lambda, y_1, \ldots, y_n\right) \\
&\propto \underbrace{\left[\prod_{l=1}^{\tilde{m}} \left(\prod_{k=1}^{2^{\tilde{m}-l}} \Phi\left\{\omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l}\right) - \omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l+2^{l-1}}\right)\right\} \right) \right]}_{\text{the first term}} \times \underbrace{\frac{1}{1+e^{-\left(\omega(y_i)-\omega\left(t^{\tilde{m}}_{i,1}\right)\right)}}}_{\text{the second term}}.
\end{aligned} \tag{3.13}$$

Therefore, for the depth-$\tilde{m}$ tree $\mathcal{T}_i^{\tilde{m}}$ associated with observation y_i, we can firstly generate proposals $\mathcal{T}_i^{\tilde{m}} = \left\{t^{\tilde{m}}_{i,1}, t^{\tilde{m}}_{i,2}, \ldots, t^{\tilde{m}}_{i,2^{\tilde{m}}}\right\}$ by running Algorithm 3. It should be noted that here we regard the first term as a proposal distribution over $\mathcal{T}_i^{\tilde{m}}$ and then decide whether to accept or reject the entire tree $\mathcal{T}_i^{\tilde{m}}$ based on the second term. If we accept, $\mathcal{T}_i^{\tilde{m}}$ will be retained and used for subsequent sampling steps; otherwise, we have to keep running Algorithm 3 until an accepted tree is obtained.

▷ **Sampling Pólya-gamma Variables**

After sampling $\mathcal{T}_i^{\tilde{m}}$, $\tilde{m} = 0, 1, \ldots, m-1$ with Algorithm 3 and rejection sampling, we sample

$\mathcal{W}_i = \{\mathcal{W}_i^{\tilde{m}}\}_{\tilde{m}=0}^{m-1}$ from its full conditional,

$$
\begin{aligned}
& p\left(\mathcal{W}_i \mid \{\mathcal{T}_{i'}\}_{i'=1}^{n}, \{\mathcal{W}_{i'}\}_{i' \neq i}, \omega, \lambda, y_1, \ldots, y_n\right) \\
\propto & \prod_{\tilde{m}=1}^{m-1}\left[\prod_{l=1}^{\tilde{m}}\left(\prod_{k=1}^{2^{\tilde{m}-l}} e^{h\left(w^{\tilde{m}}_{i,\left(1+(k-1)2^l,1+(k-1)2^l+2^{l-1}\right)},\omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l}\right)-\omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l+2^{l-1}}\right)\right)}\right.\right. \\
& \left.\left.\times\, p_{\mathrm{PG}}\left(w^{\tilde{m}}_{i,\left(1+(k-1)2^l,1+(k-1)2^l+2^{l-1}\right)} \mid 1,0\right)\right) e^{h\left(w_i^{\tilde{m}},\omega(y_i)-\omega\left(t_{i,1}^{\tilde{m}}\right)\right)} p_{\mathrm{PG}}\left(w_i^{\tilde{m}} \mid 1,0\right)\right] \\
& \times \exp\left\{h\left(w_i^0, \omega(y_i)-\omega\left(t_{i,1}^0\right)\right)\right\} p_{\mathrm{PG}}\left(w_i^0 \mid 1,0\right) \\
\propto & \prod_{\tilde{m}=1}^{m-1}\left[\prod_{l=1}^{\tilde{m}}\left(\prod_{k=1}^{2^{\tilde{m}-l}} p_{\mathrm{PG}}\left(w^{\tilde{m}}_{i,(1+k2^l,1+(k-1)2^l+2^{l-1})} \mid 1, \omega\left(t^{\tilde{m}}_{1+(k-1)2^l}\right)-\omega\left(t^{\tilde{m}}_{1+(k-1)2^l+2^{l-1}}\right)\right)\right)\right. \\
& \left.\times\, p_{\mathrm{PG}}\left(w_i^{\tilde{m}} \mid 1, \omega(y_i)-\omega\left(t_{i,1}^{\tilde{m}}\right)\right)\right] \times p_{\mathrm{PG}}\left(w_i^0 \mid 1, \omega(y_i)-\omega\left(t_{i,1}^0\right)\right)
\end{aligned}
\tag{3.14}
$$

which implies that

$$
\begin{aligned}
& p\left(\mathcal{W}_i^{\tilde{m}} \mid \{\mathcal{T}_{i'}\}_{i'=1}^{n}, \{\mathcal{W}_{i'}\}_{i' \neq i}, \omega, \lambda, y_1, \ldots, y_n\right) \propto p_{\mathrm{PG}}\left(w_i^{\tilde{m}} \mid 1, \omega(y_i)-\omega\left(t_{i,1}^{\tilde{m}}\right)\right) \\
& \times \prod_{l=1}^{\tilde{m}} \prod_{k=1}^{2^{\tilde{m}-l}} p_{\mathrm{PG}}\left(w^{\tilde{m}}_{i,\left(1+(k-1)2^l,1+(k-1)2^l+2^{l-1}\right)} \mid 1, \omega\left(t^{\tilde{m}}_{1,1+(k-1)2^l}\right)-\omega\left(t^{\tilde{m}}_{1,1+(k-1)2^l+2^{l-1}}\right)\right).
\end{aligned}
\tag{3.15}
$$

Then, we can sample $w_i^{\tilde{m}}$ from $p_{\mathrm{PG}}\left(\cdot \mid 1, \omega(y_i)-\omega\left(t_{i,1}^{\tilde{m}}\right)\right)$ and $w^{\tilde{m}}_{i,\left(1+(k-1)2^l,1+(k-1)2^l+2^{l-1}\right)}$ from $p_{\mathrm{PG}}\left(\cdot \mid 1, \omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l}\right)-\omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l+2^{l-1}}\right)\right)$.

3.2.2.2 Sampling Gaussian processes

In this part, we need to sample the values of ω at both observations $y_1, y_2, \ldots, y_n$ as well as rejected proposals $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$ conditional on the other variables. For simplicity, let $\boldsymbol{\omega} = \left(\boldsymbol{\omega}_{\mathrm{obs}}^T, \left(\boldsymbol{\omega}_1^0\right)^T, \ldots, \left(\boldsymbol{\omega}_1^{m-1}\right)^T, \left(\boldsymbol{\omega}_2^0\right)^T, \ldots, \left(\boldsymbol{\omega}_2^{m-1}\right)^T, \ldots, \left(\boldsymbol{\omega}_n^0\right)^T, \ldots, \left(\boldsymbol{\omega}_n^{m-1}\right)^T\right)^T$, where $\boldsymbol{\omega}_{\mathrm{obs}} = (\omega(y_1), \omega(y_2), \ldots, \omega(y_n))^T$, $\boldsymbol{\omega}_i^{\tilde{m}} = \left(\omega\left(t_{i,1}^{\tilde{m}}\right), \ldots, \omega\left(t_{2^{\tilde{m}}}^{\tilde{m}}\right)\right)^T$, $i = 1, 2, \ldots, n$, $\tilde{m} = 0, 1, \ldots, m-$

1. Afterwards, all we need to do is to derive the mean and covariance of the full conditional for $\boldsymbol{\omega}$, which is a multivariate Gaussian as follows,

$$
\begin{aligned}
& p\left(\boldsymbol{\omega} \mid \mathcal{T}_1, \mathcal{W}_1, \mathcal{T}_2, \mathcal{W}_2, \ldots, \mathcal{T}_n, \mathcal{W}_n, \lambda, y_1, y_2, \ldots, y_n\right) \\
& \propto \left\{ \prod_{i=1}^{n} \left[\prod_{\tilde{m}=1}^{m-1} \prod_{l=1}^{\tilde{m}} \left(\prod_{k=1}^{2^{\tilde{m}-l}} e^{h\left(w^{\tilde{m}}_{i,\left(1+(k-1)2^l, 1+(k-1)2^l+2^{l-1}\right)}, \omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l}\right) - \omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l+2^{l-1}}\right)\right)} \right) \right. \right. \\
& \left. \left. \times e^{h\left(w_i^{\tilde{m}}, \omega(y_i) - \omega\left(t^{\tilde{m}}_{i,1}\right)\right)} \right] e^{h\left(w_i^0, \omega(y_i) - \omega\left(t^0_{i,1}\right)\right)} \right\} \times \mathcal{GP}\left(\boldsymbol{\omega} \mid y_1, \ldots, y_n, \mathcal{T}_1, \ldots, \mathcal{T}_n\right) \\
& \propto \prod_{i=1}^{n} \prod_{\tilde{m}=1}^{m-1} \prod_{l=1}^{\tilde{m}} \prod_{k=1}^{2^{\tilde{m}-l}} \mathcal{GP}\left(\boldsymbol{\omega} \mid y_1, y_2, \ldots, y_n, \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n\right) e^{-\frac{\left(\omega(y_i) - \omega\left(t^{\tilde{m}}_{i,1}\right)\right)^2}{2} w_i^{\tilde{m}} + \frac{\omega(y_i) - \omega\left(t^{\tilde{m}}_{i,1}\right)}{2}} \\
& \quad \times e^{-\frac{\left(\omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l}\right) - \omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l+2^{l-1}}\right)\right)^2}{2} w^{\tilde{m}}_{i,\left(1+(k-1)2^l, 1+(k-1)2^l+2^{l-1}\right)} + \frac{\omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l}\right) - \omega\left(t^{\tilde{m}}_{i,1+(k-1)2^l+2^{l-1}}\right)}{2}} \\
& = \exp\left(-\frac{1}{2}\boldsymbol{\omega}^T \Lambda \boldsymbol{\omega} + \boldsymbol{\omega}^T c\right) \times \exp\left(-\frac{1}{2}\boldsymbol{\omega}^T K^{-1} \boldsymbol{\omega}\right) \\
& \propto \exp\left(-\frac{1}{2}(\boldsymbol{\omega} - \mu)^T \Sigma^{-1} (\boldsymbol{\omega} - \mu)\right),
\end{aligned}
\tag{3.16}
$$

where

$$
\Lambda = \begin{pmatrix}
\Lambda_{0,0} & \Lambda_{0,1} & \Lambda_{0,2} & \cdots & \Lambda_{0,n} \\
\Lambda_{1,0} & \Lambda_{1,1} & \mathbf{0} & \cdots & \mathbf{0} \\
\Lambda_{2,0} & \mathbf{0} & \Lambda_{2,2} & \cdots & \mathbf{0} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\Lambda_{n,0} & \mathbf{0} & \mathbf{0} & \cdots & \Lambda_{n,n}
\end{pmatrix}, \quad
\Lambda_{0,0} = \begin{pmatrix}
\sum_{\tilde{m}=0}^{m-1} w_1^{\tilde{m}} & 0 & \cdots & 0 \\
0 & \sum_{\tilde{m}=0}^{m-1} w_2^{\tilde{m}} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \sum_{\tilde{m}=0}^{m-1} w_n^{\tilde{m}}
\end{pmatrix},
$$

$$
\Lambda_{0,i} = \begin{pmatrix}
\mathbf{0}_{(i-1)\times(2^m-1)} \\
\lambda_i^T \\
\mathbf{0}_{(n-i)\times(2^m-1)}
\end{pmatrix}, \quad
\lambda_i = \left(-w_i^0, -w_i^1, 0, \ldots, -w_i^{\tilde{m}}, \underbrace{0, \ldots, 0}_{2^{\tilde{m}}-1}, \ldots, -w_i^{m-1}, \underbrace{0, \ldots, 0}_{2^{m-1}-1} \right)^T,
$$

$$\Lambda_{i,i} = \begin{pmatrix} \Lambda_{i,i}^0 & & & \\ & \Lambda_{i,i}^1 & & \\ & & \ddots & \\ & & & \Lambda_{i,i}^{m-1} \end{pmatrix}, \quad K = \begin{pmatrix} \sigma_\lambda(y_1,y_1) & \sigma_\lambda(y_1,y_2) & \cdots & \sigma_\lambda(y_1,y_n) \\ \sigma_\lambda(y_2,y_1) & \sigma_\lambda(y_2,y_2) & \cdots & \sigma_\lambda(y_2,y_n) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_\lambda(y_n,y_1) & \sigma_\lambda(y_n,y_2) & \cdots & \sigma_\lambda(y_n,y_n) \end{pmatrix},$$

$\Lambda_{i,i}^{\tilde{m}}$ and the vector c are related to Pólya-gamma variables introduced by the depth-$\tilde{m}$ binary tree $T_i^{\tilde{m}}$ of observation y_i, $\sigma_\lambda(u,v) = \sigma_0(\lambda u, \lambda v)$, $\sigma_0(s,t)$ is a pre-specified positive definite function, $\Sigma = (K^{-1} + \Lambda)^{-1}$ and $\mu = \Sigma c$.

3.3 The Relationship between the Processes GP and TLLGP

In order to demonstrate the weak consistency of TLLGP, we start by exploring the relationship between the processes ω and L_ω^m in an attempt to find the Kullback–Leibler support of TLLGP.

Theorem 2. *For any two functions $\omega_1(t)$, $\omega_2(t)$ on T and a probability density function $\pi(t)$ on T,*

$$\|\omega_1 - \omega_2\|_\infty < \epsilon \Rightarrow \left\| \log \frac{L_{\omega_1}\pi}{L_{\omega_2}\pi} \right\|_\infty < 2\epsilon. \tag{3.17}$$

Proof. By $\|\omega_1 - \omega_2\|_\infty < \epsilon$, we have

$$\omega_2(t) - \epsilon < \omega_1(t) < \omega_2(t) + \epsilon, \ \forall t \in T. \tag{3.18}$$

Therefore, we have

$$\begin{aligned}
&\log \frac{L_{\omega_1}\pi}{L_{\omega_2}\pi} \\
=&\log \frac{2\int_T \frac{e^{\omega_1(t)}}{e^{\omega_1(t)}+e^{\omega_1(s)}}\pi(t)\pi(s)ds}{2\int_T \frac{e^{\omega_2(t)}}{e^{\omega_2(t)}+e^{\omega_2(s)}}\pi(t)\pi(s)ds} \\
=&\log\left\{ e^{\omega_1(t)-\omega_2(t)} \frac{\int_T \frac{1}{e^{\omega_1(t)}+e^{\omega_1(s)}}\pi(t)\pi(s)ds}{\int_T \frac{1}{e^{\omega_2(t)}+e^{\omega_2(s)}}\pi(t)\pi(s)ds} \right\} \\
<&\log\left\{ e^{\omega_1(t)-\omega_2(t)} \frac{\int_T \frac{1}{e^{\omega_2(t)}e^{-\epsilon}+e^{\omega_2(s)}e^{-\epsilon}}\pi(t)\pi(s)ds}{\int_T \frac{1}{e^{\omega_2(t)}+e^{\omega_2(s)}}\pi(t)\pi(s)ds} \right\} \\
=&\omega_1(t) - \omega_2(t) + \epsilon,
\end{aligned} \tag{3.19}$$

and

$$\begin{aligned}
&\log \frac{L_{\omega_1}\pi}{L_{\omega_2}\pi} \\
=&\log \left\{ e^{\omega_1(t)-\omega_2(t)} \frac{\int_T \frac{1}{e^{\omega_1(t)}+e^{\omega_1(s)}}\pi(t)\pi(s)ds}{\int_T \frac{1}{e^{\omega_2(t)}+e^{\omega_2(s)}}\pi(t)\pi(s)ds} \right\} \\
>&\log \left\{ e^{\omega_1(t)-\omega_2(t)} \frac{\int_T \frac{1}{e^{\omega_2(t)}e^{\epsilon}+e^{\omega_2(s)}e^{\epsilon}}\pi(t)\pi(s)ds}{\int_T \frac{1}{e^{\omega_2(t)}+e^{\omega_2(s)}}\pi(t)\pi(s)ds} \right\} \\
=&\omega_1(t)-\omega_2(t)-\epsilon.
\end{aligned} \tag{3.20}$$

In sum, we have

$$\begin{aligned}
&\omega_1(t)-\omega_2(t)-\epsilon < \log \frac{L_{\omega_1}\pi}{L_{\omega_2}\pi} < \omega_1(t)-\omega_2(t)+\epsilon,\ \forall t \in T, \\
\Rightarrow &-\epsilon-\epsilon < \omega_1(t)-\omega_2(t)-\epsilon < \log \frac{L_{\omega_1}\pi}{L_{\omega_2}\pi} < \omega_1(t)-\omega_2(t)+\epsilon < \epsilon+\epsilon,\ \forall t \in T, \\
\Rightarrow &\left\| \log \frac{L_{\omega_1}\pi}{L_{\omega_2}\pi} \right\|_\infty < 2\epsilon.
\end{aligned} \tag{3.21}$$

□

Theorem 3. *For any two functions $\omega_1(t)$, $\omega_2(t)$ on T, a probability density function $\pi(t)$ on T and a pre-specified positive integer m,*

$$\|\omega_1-\omega_2\|_\infty < \epsilon \Rightarrow \left\| \log \frac{L^m_{\omega_1}\pi}{L^m_{\omega_2}\pi} \right\|_\infty < 2\left(2^m-1\right)\epsilon. \tag{3.22}$$

Proof. Below we will prove Equation 3.22 by induction.

1. When $m=1$, we can find that Equation 3.22 holds by Theorem 2.
2. Assume that for some positive integer k, the inequality holds, i.e.,

$$\|\omega_1-\omega_2\|_\infty < \epsilon \Rightarrow \left\| \log \frac{L^k_{\omega_1}\pi}{L^k_{\omega_2}\pi} \right\|_\infty < 2\left(2^k-1\right)\epsilon, \tag{3.23}$$

which implies that

$$\begin{aligned} &-2\left(2^k-1\right)\epsilon < \log\frac{L^k_{\omega_1}(t)}{L^k_{\omega_2}(t)} < 2\left(2^k-1\right)\epsilon \\ \Rightarrow &e^{-2\left(2^k-1\right)\epsilon}\left(L^k_{\omega_2}\right)(t) < \left(L^k_{\omega_1}\right)(t) < e^{2\left(2^k-1\right)\epsilon}\left(L^k_{\omega_2}\right)(t),\ \forall t\in T. \end{aligned} \tag{3.24}$$

Then, we only need to prove that Equation 3.22 holds for $k+1$ as well. Similar to what we've done for Theorem 2, we have

$$\begin{aligned} &\log\frac{L^{k+1}_{\omega_1}\pi}{L^{k+1}_{\omega_2}\pi} \\ =&\log\frac{L_{\omega_1}\left(L^k_{\omega_1}\pi\right)}{L_{\omega_2}\left(L^k_{\omega_2}\pi\right)} \\ =&\log\frac{2\int_T \frac{e^{\omega_1(t)}}{e^{\omega_1(t)}+e^{\omega_1(s)}}\left(L^k_{\omega_1}\pi\right)(t)\left(L^k_{\omega_1}\pi\right)(s)ds}{2\int_T \frac{e^{\omega_2(t)}}{e^{\omega_2(t)}+e^{\omega_2(s)}}\left(L^k_{\omega_2}\pi\right)(t)\left(L^k_{\omega_2}\pi\right)(s)ds} \\ =&\log\left\{e^{\omega_1(t)-\omega_2(t)}\frac{\int_T \frac{1}{e^{\omega_1(t)}+e^{\omega_1(s)}}\left(L^k_{\omega_1}\pi\right)(t)\left(L^k_{\omega_1}\pi\right)(s)ds}{\int_T \frac{1}{e^{\omega_2(t)}+e^{\omega_2(s)}}\left(L^k_{\omega_2}\pi\right)(t)\left(L^k_{\omega_2}\pi\right)(s)ds}\right\} \\ <&\log\left\{e^{\omega_1(t)-\omega_2(t)}\frac{\int_T \frac{1}{e^{\omega_2(t)}e^{-\epsilon}+e^{\omega_2(s)}e^{-\epsilon}}e^{2\left(2^k-1\right)\epsilon}\left(L^k_{\omega_2}\pi\right)(t)e^{2\left(2^k-1\right)\epsilon}\left(L^k_{\omega_2}\pi\right)(s)ds}{\int_T \frac{1}{e^{\omega_2(t)}+e^{\omega_2(s)}}\left(L^k_{\omega_2}\pi\right)(t)\left(L^k_{\omega_2}\pi\right)(s)ds}\right\} \\ =&\omega_1(t)-\omega_2(t)+\epsilon+2\left(2^k-1\right)\epsilon+2\left(2^k-1\right)\epsilon, \end{aligned} \tag{3.25}$$

and

$$
\begin{aligned}
&\log \frac{L_{\omega_1}^{k+1}\pi}{L_{\omega_2}^{k+1}\pi} \\
=&\log \left\{ e^{\omega_1(t)-\omega_2(t)} \frac{\int_T \frac{1}{e^{\omega_1(t)}+e^{\omega_1(s)}} \left(L_{\omega_1}^{k}\pi\right)(t) \left(L_{\omega_1}^{k}\pi\right)(s) ds}{\int_T \frac{1}{e^{\omega_2(t)}+e^{\omega_2(s)}} \left(L_{\omega_2}^{k}\pi\right)(t) \left(L_{\omega_2}^{k}\pi\right)(s) ds} \right\} \\
>&\log \left\{ e^{\omega_1(t)-\omega_2(t)} \frac{\int_T \frac{1}{e^{\omega_2(t)}e^{\epsilon}+e^{\omega_2(s)}e^{\epsilon}} e^{-2\left(2^k-1\right)\epsilon} \left(L_{\omega_2}^{k}\pi\right)(t) e^{-2\left(2^k-1\right)\epsilon} \left(L_{\omega_2}^{k}\pi\right)(s) ds}{\int_T \frac{1}{e^{\omega_2(t)}+e^{\omega_2(s)}} \left(L_{\omega_2}^{k}\pi\right)(t) \left(L_{\omega_2}^{k}\pi\right)(s) ds} \right\} \\
=&\omega_1(t)-\omega_2(t)-\epsilon-2\left(2^k-1\right)\epsilon-2\left(2^k-1\right)\epsilon.
\end{aligned}
\tag{3.26}
$$

$\forall t \in T$, we have

$$
\begin{aligned}
&(\omega_1-\omega_2)(t)-\epsilon-2^2\left(2^k-1\right)\epsilon < \log \frac{L_{\omega_1}^{k+1}\pi}{L_{\omega_2}^{k+1}\pi} < (\omega_1-\omega_2)(t)+\epsilon+2^2\left(2^k-1\right)\epsilon, \\
\Rightarrow &-2\epsilon-2^2\left(2^k-1\right)\epsilon < \log \frac{L_{\omega_1}^{k+1}\pi}{L_{\omega_2}^{k+1}\pi} < 2\epsilon+2^2\left(2^k-1\right)\epsilon,
\end{aligned}
\tag{3.27}
$$

which implies

$$
\left\| \log \frac{L_{\omega_1}^{k+1}\pi}{L_{\omega_2}^{k+1}\pi} \right\|_{\infty} < 2\left(2^{k+1}-1\right)\epsilon. \tag{3.28}
$$

In sum, by the principle of mathematical induction, Equation 3.22 holds. □

Then, by Theorem 3, we have

$$
\begin{aligned}
&\|\omega_1-\omega_2\|_{\infty} < \epsilon \\
\Rightarrow &\int L_{\omega_1}^{m}\pi \frac{L_{\omega_1}^{m}\pi}{L_{\omega_2}^{m}\pi} < \int L_{\omega_1}^{m}\pi \left| \frac{L_{\omega_1}^{m}\pi}{L_{\omega_2}^{m}\pi} \right| < 2\left(2^m-1\right)\epsilon.
\end{aligned}
\tag{3.29}
$$

4. Numerical Experiments

In this chapter, we carry out comprehensive numerical experiments to evaluate different models' performances in density estimation.

4.1 A Low-rank Approximation

Before talking about details of our numerical experiments, we would like to introduce discretization of the GP mentioned in Chapter 1. For efficient computing, we use a finite-dimensional approximation to the Gaussian process (GP). We retain the values of ω only at a finite set of nodes $\{\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_r\}$, and impute the rest with the conditional expectation. Formally, we approximate ω by a predictive process $\tilde{\omega}(t) = \mathbb{E}[\omega(t) \mid \boldsymbol{\omega}_r, \lambda] = \tilde{\sigma}_\lambda^T(t)\tilde{\Sigma}_\lambda^{-1}\boldsymbol{\omega}_r = A_\lambda^T(t)Z,\ t \in T$, where $A_\lambda(t) = \tilde{\Sigma}_\lambda^{-1/2}\tilde{\sigma}_\lambda(t)$,

$$\tilde{\sigma}_\lambda(t) = \begin{pmatrix} \sigma_\lambda(\tilde{t}_1, t) \\ \sigma_\lambda(\tilde{t}_2, t) \\ \vdots \\ \sigma_\lambda(\tilde{t}_r, t) \end{pmatrix}, \tilde{\Sigma}_\lambda = \begin{pmatrix} \sigma_\lambda(\tilde{t}_1, \tilde{t}_1) & \sigma_\lambda(\tilde{t}_1, \tilde{t}_2) & \cdots & \sigma_\lambda(\tilde{t}_1, \tilde{t}_r) \\ \sigma_\lambda(\tilde{t}_2, \tilde{t}_1) & \sigma_\lambda(\tilde{t}_2, \tilde{t}_2) & \cdots & \sigma_\lambda(\tilde{t}_2, \tilde{t}_r) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_\lambda(\tilde{t}_r, \tilde{t}_1) & \sigma_\lambda(\tilde{t}_r, \tilde{t}_2) & \cdots & \sigma_\lambda(\tilde{t}_r, \tilde{t}_r) \end{pmatrix}.$$

The conditional expectation implies that when λ is given, $\tilde{\omega}$ has the same distribution as the process $A_\lambda^T(\cdot)Z$, where $Z \sim \mathcal{N}_r(0, I_r)$.

4.2 The Setup

To evaluate the performances of our approach, we generated simulated data sets from the three underlying distributions f_0 listed in Table 4.1. To provide a more intuitive understanding, we draw their probability density functions in Figure 4.1. From each distribution, we generated 10 data sets with a sample size of 100 each.

Then, we applied four models for the density estimation task, including logistic Gaussian process (LGP) in Chapter 1, logistic-link Gaussian processes (LLGP) in Chapter 2 as well as tree-logistic-link Gaussian processes (TLLGP) in Chapter 3 with depth 3 and 5. For the sake of simplicity and practicality, we won't update the models' inverse length-scale λ during the sampling processes. Instead, for data sets generated from three underlying distributions, we set each model's $\lambda = 10$, which is implied by cross-validation. As for the

Table 4.1: List of true f_0 used in our numerical experiments to assess how large the prior support is. Here $\mathsf{Unif}(l, u)$ is the uniform distribution over the interval (l, u), $\mathsf{Be}(a, b)$ is the beta distribution with shapes a and b, $\mathsf{tEx}(\lambda)$ is the exponential distribution with rate λ truncated to the unit interval.

Name	f_0
Hump	$0.75 \cdot \mathsf{tEx}(3) + 0.2 \cdot \mathsf{Be}(12, 8) + 0.05 \cdot \mathsf{Unif}(0, 1)$
Peaks	$0.4 \cdot \mathsf{Be}(18, 138) + 0.2 \cdot \mathsf{Be}(90, 30) + 0.3 \cdot \mathsf{Be}(30, 30) + 0.1 \cdot \mathsf{Unif}(0, 1)$
Mixed	$0.6 \cdot \mathsf{Be}(18, 138) + 0.3 \cdot \mathsf{Be}(10, 10) + 0.1 \cdot \mathsf{Unif}(0, 1)$

variance parameter κ^2 in each model's covariance function, we updated it in LGP and LLGP. It should be noted that the priors for κ^2 we used in LGP and LLGP are an inverse-gamma distribution with shape parameter and rate parameter both being $3/2$ (Tokdar et al., 2024) and a half-Cauchy distribution with location being 0 and scale being 1, separately. For TLLGP, the scale of density curves are mainly controlled by the tree depth. As a result, we set $\kappa^2 = 1$ and let the tree depth be 3 and 5, respectively.

Lastly, as we have mentioned in Section 4.1, for efficient posterior computation, we apply a low-rank approximation to the original GP. Here the knots used in our numerical experiments are $\{0.0, 0.1, 0.2, \ldots, 0.9, 1.0\}$; the grids used for numerical integration are $\{0.00, 0.01, 0.02, \ldots, 0.99, 1.00\}$. We ran $3,000$ iterations each for four models, and then discarded the first $1,000$ draws as burn-in.

4.3 Evaluation

Firstly, we evaluate the performances of the four models in density estimation. Specifically, we adopt the metric integrated mean squared error (IMSE) to compare their performances which are summarized in Table 4.2. The IMSE is defined as

$$\mathrm{IMSE}(f_0, \hat{f}) = \int_T \left[f_0(s) - \hat{f}(s) \right]^2 ds, \tag{4.1}$$

where $f_0(t)$ is the true probability density function and $\hat{f}(t)$ is the estimated probability density function from observations.

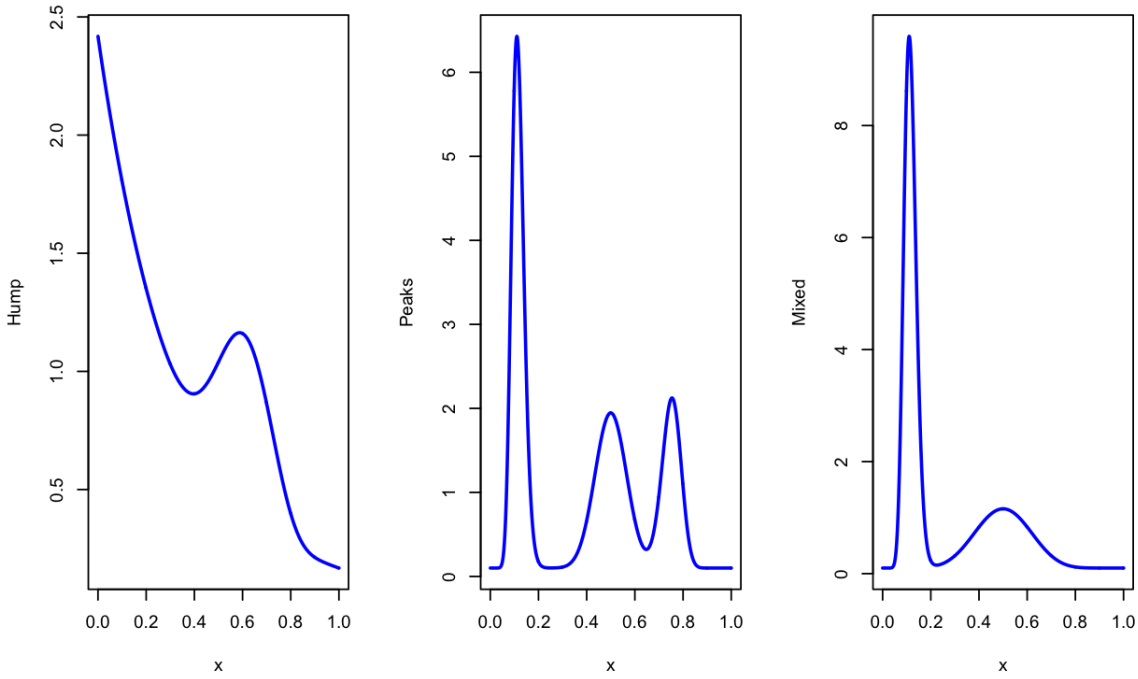

FIGURE 4.1: Probability density functions for three distributions used in our numerical experiments.

Table 4.2: IMSE of four density models on data sets generated from three distributions.

Distributions	LGP	LLGP	TLLGP with depth 3	TLLGP with depth 5
Hump	0.0481	0.0773	0.0820	0.0935
Peaks	0.6534	0.2182	0.3191	0.1711
Mixed	0.4148	0.1818	0.8616	0.2175

Apart from density estimation, we also care out these density models' performances in terms of mixing time and run-time. Here we compare these models' mixing time using effective sample size (ESS). It should be noted that in our problem, the model parameters are density functions over $[0, 1]$. Therefore, we only present ESS of the estimated densities at the given knots $\{0.0, 0.1, 0.2, \ldots, 0.9, 1.0\}$ in Tables 4.3, 4.4 and 4.5.

Table 4.3: ESS of estimated densities at the knots when adopting four density models on data sets generated from **Hump**.

Knots	LGP	LLGP	TLLGP with depth 3	TLLGP with depth 5
0.0	61.999	332.857	859.381	664.994
0.1	67.705	341.897	903.368	612.413
0.2	64.382	523.895	903.728	548.150
0.3	63.099	542.107	867.318	494.980
0.4	62.827	615.637	816.671	513.711
0.5	66.192	527.735	883.390	491.715
0.6	72.258	628.871	847.378	508.837
0.7	75.093	621.368	878.561	462.027
0.8	65.364	677.594	667.868	351.490
0.9	59.574	383.974	504.964	244.520
1.0	64.879	575.519	575.270	255.225

Table 4.4: ESS of estimated densities at the knots when adopting four density models on data sets generated from **Peaks**.

Knots	LGP	LLGP	TLLGP with depth 3	TLLGP with depth 5
0.0	49.505	297.013	610.857	297.448
0.1	45.669	108.443	522.252	448.686
0.2	48.746	212.891	471.311	232.702
0.3	49.041	222.859	344.148	149.817
0.4	59.429	443.436	573.054	314.029

0.5	50.129	534.381	642.059	465.078
0.6	49.420	430.257	571.906	356.862
0.7	66.449	476.127	622.476	361.433
0.8	48.772	533.020	601.023	419.706
0.9	60.923	206.490	361.117	164.323
1.0	148.078	278.210	441.762	172.260

Table 4.5: ESS of estimated densities at the knots when adopting four density models on data sets generated from **Mixed**.

Knots	**LGP**	**LLGP**	**TLLGP with depth 3**	**TLLGP with depth 5**
0.0	49.354	122.993	448.058	252.214
0.1	56.433	113.910	383.694	382.578
0.2	40.807	154.530	408.549	246.479
0.3	58.468	196.758	415.836	236.602
0.4	65.138	332.684	548.972	357.974
0.5	65.512	364.407	528.034	389.316
0.6	66.023	319.807	513.517	348.973
0.7	74.918	190.397	440.752	263.789
0.8	74.764	110.964	281.125	132.480
0.9	90.353	85.113	289.667	117.085
1.0	122.250	105.60	379.958	163.353

Lastly, For practical matters, we give the run-time of density models except LGP in Table 4.6, and all times are recorded in minutes.

Table 4.6: The run-time of four density models on data sets generated from three distributions.

Distributions	LLGP	TLLGP with depth 3	TLLGP with depth 5
Hump	19.97	21.53	52.95
Peaks	17.82	19.26	49.62
Mixed	14.07	19.25	48.29

5. Discussion

The numerical experiments in Chapter 4 evaluate different aspects of four density models in terms of density estimation (IMSE), mixing time of the Markov chain (ESS) as well as run-time.

When the density function of the underlying distribution varies slowly like **Hump** used in our numerical experiments, LGP performs the best in density estimation. For such types of densities functions, LLGP as well as TLLGP with depth 3 and 5 have similar performances. Admittedly, since the sample size of our simulated data sets is small, adopting a complex density model, such as TLLGP with depth 5, could easily get stuck in the overfitting problem. On the contrary, TLLGP with a deeper generative structure has much better performance than other simple models such as TLLGP with depth 3 and LGP. As we have shown in Chapter 4, the probability density function of **Peaks** varies rapidly in certain regions and has multiple modes. In such cases, using a TLLGP with more layers would be more appropriate. Then, based on the results for **Mixed**, we again demonstrate the advantage of adding more layers in TLLGP. For observations generated from **Mixed**, TLLGP with depth 3 performs the worst, which is consistent with our expectations, since the peak of the density function of this distribution has already exceeded 8, while TLLGP can only fit density functions f with $\|f\|_\infty < 2^3$.

For all three distributions, it is obvious that the mixing of the estimated density functions' values by TLLGP at the given knots are better than the others. Specifically, TLLGP with depth 3 has better mixing, since it's easier for TLLGP with fewer layers to get rid of regions corresponding to lots of rejections while sampling. This is also why LLGP is mixing worse than TLLGP with depth 5 at most knots. Due to the presence of the integral over the whole sample path of GP, LGP is mixing worst among these density models, which further demonstrates the advantage of the data augmentation techniques we introduced.

Finally, as for the actual running time of each density model, TLLP with depth 5 clearly has the longest time due to its deeper hierarchical structure, which sometimes requires a significant amount of time to sample and obtain a complete accepted tree with depth 4.

TLLGP with depth 3 and LLGP have similar and comparable running times. Given that the posterior sampling procedure for TLLGP requires further optimization, we believe that the TLLGP program, after improvements, will be able to complete posterior computation within a shorter amount of time.

6. Conclusion

In this thesis, we have developed and thoroughly examined the tree-logistic-link Gaussian process (TLLGP), a new approach for nonparametric Bayesian density estimation. The innovation lies in its ability to significantly reduce computational time without sacrificing the accuracy and flexibility inherent in Gaussian process (GP) models. Through a generative process represented by a binary tree structure, TLLGP allows for more efficient posterior computations, addressing one of the main limitations of the logistic Gaussian Process (LGP).

Our experiments confirm that TLLGP not only outperforms other methods in computational efficiency but also maintains competitive accuracy in density estimation. These findings suggest that TLLGP is a valuable in applications involving large multivariate data.

Furthermore, this work opens several avenues for future research, including exploring data-driven approach for selecting the tree depth, learning the inverse-length scale from observations, extending the model to high-dimensional data, and applying the TLLGP framework to other types of statistical problems, such as regression, classification, etc. By pushing the boundaries of computational efficiency and model flexibility, this thesis contributes to advancing the field of Bayesian nonparametric methods and their practical applications.

Bibliography

Adams, R. P., Murray, I., & MacKay, D. J. (2009). Nonparametric bayesian density modeling with gaussian processes. *arXiv preprint arXiv:0912.4896*.

Albert, J. H., & Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, *88*(422), 669–679.

Donner, C., & Opper, M. (2018). Efficient bayesian inference for a gaussian process density model. *arXiv preprint arXiv:1805.11494*.

Geman, S., & Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 721–741.

Lenk, P. J. (1988). The logistic normal distribution for bayesian, nonparametric, predictive densities. *Journal of the American Statistical Association*, *83*(402), 509–516.

Lenk, P. J. (1991). Towards a practicable bayesian nonparametric density estimator. *Biometrika*, *78*(3), 531–543.

Leonard, T. (1978). Density estimation, stochastic processes and prior information. *Journal of the Royal Statistical Society: Series B (Methodological)*, *40*(2), 113–132.

Li, C., Lin, L., & Dunson, D. B. (2019). On posterior consistency of tail index for bayesian kernel mixture models.

Polson, N. G., Scott, J. G., & Windle, J. (2013). Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American statistical Association*, *108*(504), 1339–1349.

Rao, V., Lin, L., & Dunson, D. B. (2016). Data augmentation for models based on rejection sampling. *Biometrika*, *103*(2), 319–335.

Tokdar, S. T. (2007). Towards a faster implementation of density estimation with logistic gaussian process priors. *Journal of Computational and Graphical Statistics*, *16*(3), 633–655.

Tokdar, S. T., Jiang, S., & Cunningham, E. L. (2022). Heavy-tailed density estimation. *Journal of the American Statistical Association*, 1–13.

Tokdar, S. T., Jiang, S., & Cunningham, E. L. (2024). Heavy-tailed density estimation. *Journal of the American Statistical Association*, *119*(545), 163–175.

Tokdar, S. T., Zhu, Y. M., & Ghosh, J. K. (2010). Bayesian density regression with logistic gaussian process and subspace projection.

Van der Vaart, A. W., & Van Zanten, J. H. (2009). Adaptive bayesian estimation using a gaussian random field with inverse gamma bandwidth.

Williams, C. K., & Rasmussen, C. E. (2006). *Gaussian processes for machine learning* (Vol. 2). MIT press Cambridge, MA.