Monotonization of Gaussian Processes

Project Report

DS3613: Semester Project

Full Name	Enrollment No.
Kartik Khurana	20221136

Supervisor : Prof. Bedartha Goswami

Department of Data Science
Indian Institute of Science Education and Research Pune
Pune, India

April 16, 2025

Contents

1 | Abstract

In paleoclimatology, constructing accurate time series from proxy records requires precise age-depth modeling. These relationships are inherently monotonic, yet standard Gaussian Processes (GPs) do not enforce such constraints. This project implements the method proposed by Riihimäki and Vehtari (2010), which introduces monotonicity into GPs using virtual derivative observations and Expectation Propagation (EP). The model [1] is built using the GPy Python library and tested on synthetic data. Results show that the method effectively incorporates prior knowledge about monotonic trends while maintaining the flexibility of nonparametric GP models.

2 | Introduction

Understanding how climate variables have changed over time is crucial for placing modern climate variability into a broader historical context. In paleoclimatology, this is achieved by analyzing proxy records—natural archives like speleothems, ice cores, or lake sediments—which store past environmental information in a stratigraphic (depth-wise) order. These records must first be converted from the depth domain to the time domain through a process known as age-depth modeling. However, this modeling is inherently uncertain due to limitations in dating methods, sparsity of dated points, and measurement errors. Accurately propagating these uncertainties into the final proxy time series remains a core challenge [3].

One crucial observation is that the depth-time relationship in most natural archives is expected to be monotonically increasing—that is, deeper layers are older. This inspired the central idea of my project: to model the age-depth relationship using Gaussian Processes (GPs) under monotonicity constraints. While standard GPs offer flexible, nonparametric regression, they do not enforce such shape constraints directly.

In Figure 2.12.1, we clearly see that there are a lot of points that violate monotonicity, i.e. where we see older sediments occurring above a younger sediment.

To address this, I implemented the method proposed by Riihimäki and Vehtari (2010) [5], which introduces monotonicity information into Gaussian Process models via virtual derivative observations. These derivative observations are not real data but probabilistic constraints inserted at selected points, encouraging the model to behave

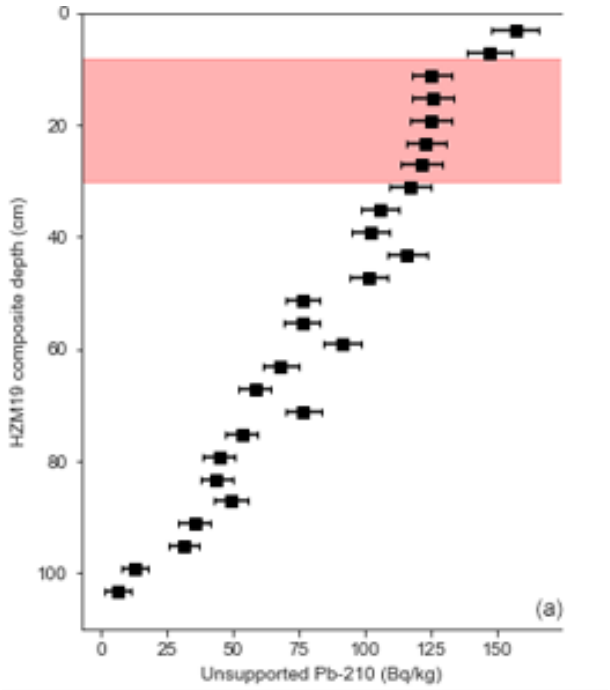

Figure 2.1: Age-Depth modeling in a sediment profile from Holzmaz, Germany [2]

monotonically in desired input dimensions. Since the resulting posterior becomes analytically intractable, Expectation Propagation (EP) is used to approximate the posterior by iteratively refining Gaussian site approximations.
The practical implementation of this method was carried out in Python using the GPy library and tested on synthetic data. While previous tools like COPRA[3] and StalAge address uncertainty in age models using Monte Carlo techniques, this GP-based approach offers a probabilistic and differentiable framework to incorporate both observational data and prior structural knowledge such as monotonicity—paving the way for more consistent proxy reconstructions.

3 | Objectives

This experiment aims to achieve the following objectives:

- To incorporate monotonicity constraints into GP models using virtual derivative observations.
- To reproduce and study the method proposed by Riihimäki & Vehtari (2010) for monotonic GPs.
- To compare standard and monotonic GP behavior through synthetic examples.

4 | Theoretical Background

Gaussian Process Regression

Gaussian Processes (GPs) provide a principled, non-parametric Bayesian approach to regression. They model distributions over functions, offering both a predictive mean and an uncertainty estimate at each test point. Formally, a GP is a collection of random variables, any finite subset of which follows a multivariate Gaussian distribution. A GP is fully specified by its mean function $m(x) = \mathbb{E}[f(x)]$ and covariance function (or kernel) $k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$ [4].
The choice of covariance function (also known as the kernel) defines the properties of functions drawn from the GP. For instance, the squared exponential (or RBF) kernel encodes the assumption that the function is infinitely differentiable and smooth (which is an requirement for having a monotone, which can be used for our desired analysis). It is given by

$$k(x, x') = \eta^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right),$$

where η^2 controls the signal variance and ℓ the length scale. The GP posterior is derived using Bayes' theorem, combining the prior and the data likelihood. Given training inputs X and corresponding noisy observations y, the GP posterior over function values at new inputs yields point predictions and also uncertainty estimates [1iteras. Again, this is a very important property with respect to our original problem because radiometric dating has high fluctuations possible over the mean value and methods like linear regression would not be able to account for that.

Modeling Derivatives in Gaussian processes

One of the appealing properties of GPs is that they are closed under linear operators and hence under differentiation as well. This means the joint distribution of function values and their derivatives is still Gaussian. As a result, one can model and make probabilistic statements about $f(x)$ and also its derivatives $f'(x)$, $f''(x)$, and so on.
The covariance between function values and derivatives, assuming the squared exponential kernel, is given by:

$$\mathrm{Cov}\left(f(x), \frac{\partial f(x')}{\partial x'}\right) = -\frac{(x - x')}{\ell^2} k(x, x')$$

$$\text{Cov}\left(\frac{\partial f(x)}{\partial x}, \frac{\partial f(x')}{\partial x'}\right) = \left(\frac{(x-x')^2}{\ell^4} - \frac{1}{\ell^2}\right) k(x, x')$$

These expressions allow us to define a joint Gaussian prior over function values and their derivatives, forming the basis for constrained GP modeling [4].

Monotonicity Constraints via Virtual Derivatives

To encode monotonicity (i.e., that $f'(x) \geq 0$ everywhere), we introduce *virtual derivative observations*. These are not real data points, but locations $x^{(i)}$ where we impose a probabilistic preference for the derivative being positive. This is done by defining a likelihood over the derivative:

$$p\left(m^{(i)} = 1 \mid \frac{\partial f}{\partial x^{(i)}}\right) = \Phi\left(\frac{1}{\nu} \cdot \frac{\partial f}{\partial x^{(i)}}\right)$$

Here $\Phi(\cdot)$ is the standard normal cumulative distribution function, and it ν is a small positive scalar controlling the softness of the constraint. As $\nu \to 0$, this likelihood approaches a step function.
Because the virtual derivative observations introduce non-Gaussian likelihood terms, the posterior becomes analytically intractable. We therefore resort to approximate inference [1iterihi.

Expectation Propagation for Inference

Expectation Propagation (EP) is an iterative algorithm that approximates non-Gaussian likelihood terms using Gaussian factors. For each virtual derivative observation, we introduce a site function:

$$t_i(f_i') \approx \mathcal{N}(f_i' \mid \hat{\mu}_i, \hat{\sigma}_i^2)$$

The EP procedure involves the following steps [4]:

1. Compute the **cavity distribution** for each site by removing the effect of the current site from the full approximation:

$$q^{-i}(f_i') = \mathcal{N}(f_i' \mid \mu_i^{-i}, (\sigma_i^{-i})^2)$$

2. Combine the cavity with the true likelihood $p(m^{(i)} \mid f_i')$ and compute the first and second moments of the resulting distribution:

$$Z_i = \int \Phi\left(\frac{1}{\nu} f_i'\right) q^{-i}(f_i')\, df_i'$$

$$\mathbb{E}[f_i'] = \mu_i^{-i} + \frac{(\sigma_i^{-i})^2}{\sqrt{(\sigma_i^{-i})^2 + \nu^2}} \cdot \frac{\phi(z_i)}{\Phi(z_i)}$$

$$\text{Var}[f_i'] = (\sigma_i^{-i})^2 - \left(\frac{(\sigma_i^{-i})^2}{(\sigma_i^{-i})^2 + \nu^2}\right) \cdot \left(z_i + \frac{\phi(z_i)}{\Phi(z_i)}\right) \cdot \frac{\phi(z_i)}{\Phi(z_i)}$$

where $z_i = \mu_i^{-i} / \sqrt{(\sigma_i^{-i})^2 + \nu^2}$, $\phi(z)$ is the standard normal PDF.

3. Update the site parameters (τ_i, ν_i) based on the difference between the new moments and the cavity distribution:

$$\tau_i = \frac{1}{\text{Var}[f_i']} - \frac{1}{(\sigma_i^{-i})^2}, \quad \nu_i = \frac{\mathbb{E}[f_i']}{\text{Var}[f_i']} - \frac{\mu_i^{-i}}{(\sigma_i^{-i})^2}$$

These site parameters are then re-incorporated into the approximate posterior. This process is repeated iteratively until convergence [4].
The final GP posterior then combines real observations and these virtual derivative observations, resulting in predictions that have the monotonicity assumption while preserving uncertainty quantification.

5 | Procedure

The following steps were implemented to apply Gaussian Process regression with monotonicity information, as described by Riihimäki and Vehtari [5]:

1. **Generate Synthetic Data:** A target function was and Gaussian noise was added to simulate realistic observations. Inputs X were sampled uniformly in the interval $[0, \pi/2]$.

2. **Construct Standard GP Model:** A baseline Gaussian Process model was built using the squared exponential (RBF) kernel. This model was trained on the noisy data to produce unconstrained predictions.

3. **Define Monotonicity Directions:** A monotonicity constraint was enforced in the positive X-direction by specifying a vector $\texttt{nvd} = [1.0]$, indicating that the function is expected to be non-decreasing.

4. **Introduce Virtual Derivative Points:** Virtual inputs X_v were selected by sampling or k-means clustering over the input space. These points represent locations where monotonicity is encouraged via derivative observations.

5. **Construct Composite Likelihood:** A custom likelihood function was created that combines the Gaussian likelihood for real observations with a Probit likelihood for virtual derivative observations. This composite likelihood was attached to the GP model.

6. **Augment Training Data:** The dataset was augmented by stacking real observations and virtual derivative observations. Virtual targets were set +1 to reflect a preference for increasing slope.

7. **Optimize Model:** The model's hyperparameters (kernel lengthscale, variance, noise level) were optimized using gradient-based optimization over the augmented data.

8. **Iterative Refinement:** If the predictive gradients violated the monotonicity assumption (i.e., predicted derivative ¡ threshold), additional virtual points were added at the most violating locations. This loop was repeated until monotonicity violations disappeared or a maximum number of iterations was reached.

9. **Prediction and Visualization:** Both the monotonic and standard GP models were used to predict the function over a test grid. Mean functions and confidence intervals were plotted to compare their behaviors.

6 | Code Availability

The full implementation of the monotonic Gaussian Process models is available at:
https://github.com/Kartik-0-0-9/Monotonization-of-Gaussian-Processes.git

7 | Results

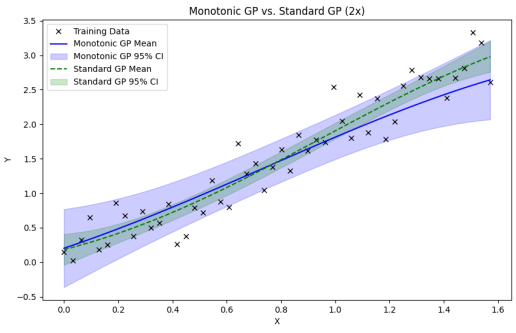

Figure 7.1: Y = 2X

Observation: The monotonic GP tightly fits the true function and maintains narrow confidence intervals. Standard GP is more uncertain and smooths over sharp trends.
Explanation for Report: Under reduced observation noise, the monotonic GP more effectively leverages the prior knowledge of increasing behavior. As [5] describes, virtual derivative observations enforce the prior over function derivatives, constraining posterior samples to respect monotonicity. This allows the model to confidently interpolate between data points, reducing posterior variance compared to a standard GP, which lacks such structural priors.

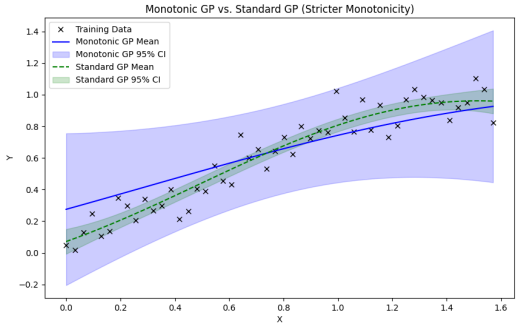

Figure 7.2: Y = sin(X) ; X ∈ $[0, \frac{\pi}{2}]$

Observation: The monotonic GP fails to fit the non-monotonic sine function, showing biased and flattened behavior, while the standard GP tracks the function's oscillations.
Explanation for Report:
This plot illustrates the impact of imposing an incorrect monotonicity prior. While the standard GP remains flexible, the monotonic GP is structurally restricted due to the derivative-based virtual observations. As expected, this leads to a poor fit when the target function violates monotonicity — a direct consequence of the prior forcing strictly positive derivatives throughout the domain, as formulated by [5] using an EP framework. The model's inability to represent local decreases leads to systematic bias and underfitting.

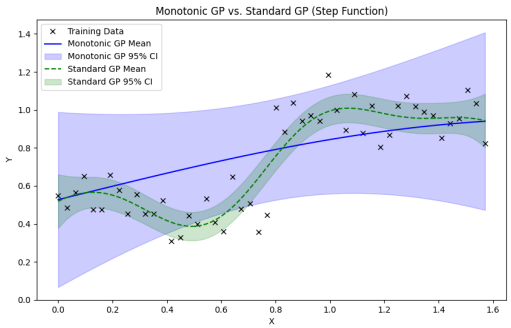

Figure 7.3: Y = 0.5 if X ¡ $\frac{\pi}{4}$; else, 1

Observation: The monotonic GP adapts to the step function's increasing behavior, capturing the sharp rise more faithfully than the standard GP,

which smooths the transition excessively.

Explanation for Report:

The monotonic GP leverages the prior that function values increase, allowing it to allocate model capacity to representing abrupt changes while avoiding over-smoothing. Standard GPs,by contrast, apply smooth kernels globally, leading to blurred transitions and poor reconstruction of discontinuities. The use of virtual derivatives in the monotonic GP, as per [5], enforces local constraints that help recover sharp upward jumps, improving fidelity without sacrificing uncertainty quantification.

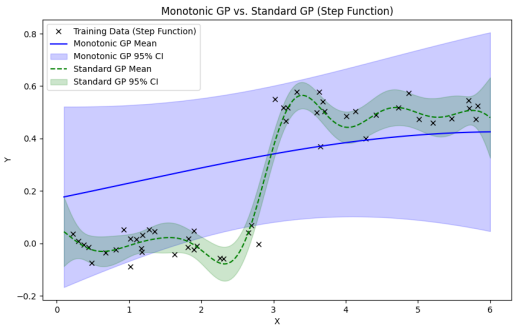

Figure 7.4: Y = 0 if X ¡ 3 ; else, 0.5 hence a 'bigger' step function

Observation: The monotonic GP fails to capture the sharp transition in the step function and produces an overly smoothed increasing trend. In contrast, the standard GP tracks the non-monotonic variations in the data, including local dips and peaks, but with increased variance near the discontinuities.

Explanation for Report: This plot highlights the trade-off between prior structural assumptions and model flexibility. The monotonic GP enforces a globally increasing trend via derivative-based constraints, which suppresses its ability to model abrupt step-like transitions. As a result, it smooths over sharp jumps, leading to bias in regions with non-monotonic behavior. The standard GP, being unconstrained, fits the data more closely, adapting to the local irregularities, but at the cost of higher uncertainty and potential overfitting near discontinuities. This example underscores the limitation of imposing monotonicity in settings where the assumption is violated, reinforcing the importance of prior compatibility with data characteristics.

Observation (for 7.5): The monotonic GP shows consistently tighter confidence intervals, smoother increasing trends, and better alignment with the ground truth in monotonic regions. In contrast, the standard GP shows more variance and sometimes overshoots or underfits in regions

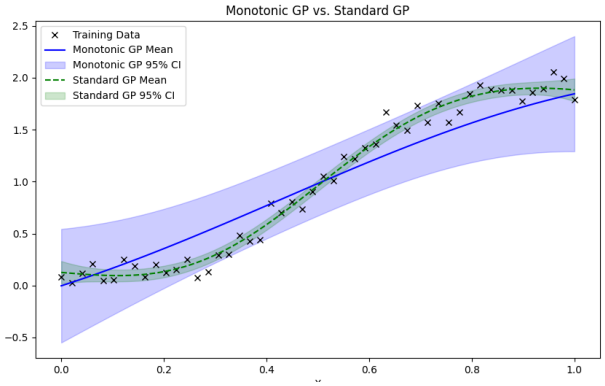

Figure 7.5: $Y = \frac{2}{1+e^{(-8X+4)}}$

with sharp transitions.

Explanation for Report:

This plot highlights the performance difference between a monotonic and standard GP across a range of input values. The monotonic GP consistently respects the underlying increasing trend, producing a smoother and more accurate fit with lower uncertainty. This is a direct outcome of incorporating prior knowledge through virtual derivative observations, as outlined by Riihimäki and Vehtari (2010). These derivative constraints reduce the flexibility of the GP in a beneficial way — eliminating non-monotonic trajectories and enforcing shape constraints. The standard GP, having no such structural prior, is prone to overfitting in regions with sparse data and over-smoothing at sharp transitions. The observed behavior confirms that monotonicity priors not only improve fit but also lead to more calibrated and confident predictions when the assumption holds.

8 | Potential Improvements

While the implemented model successfully integrates monotonicity constraints via virtual derivative observations, there are several avenues for improvement:

Constraint Softness: The strict setting of the monotonicity tolerance ($\nu = 10^{-6}$) enforces nearly hard constraints. Increasing ν would allow the model more flexibility and better robustness to noise.

Gradient Computation Efficiency: The model estimates predictive gradients using finite difference methods, which can be inefficient for high-dimensional data. Using analytical gradients or automatic differentiation libraries (e.g., Autograd or JAX) would improve scalability.

9 | Analysis and Discussion

10 | Discussion

This work explores the application of monotonic Gaussian Processes (GPs) in scenarios where the underlying function is known or assumed to be monotonic. The key motivation, following Riihimaki and Vehtari [5], is that enforcing monotonicity through virtual derivative observations can significantly improve predictive performance in terms of accuracy, uncertainty quantification, and shape conformity. Across multiple experiments, we compare standard GPs to monotonic GPs and discuss the observed behaviors in light of theoretical expectations.

10.1 | Monotonicity and Predictive Behavior

The central theoretical insight from Riihimäki and Vehtari is that monotonicity constraints introduce soft derivative information at selected points in the domain. These constraints shape the posterior distribution of the GP by reducing its flexibility to deviate from a strictly increasing or decreasing function. The effect is a model that generalizes better in regions of sparse data and avoids pathological fluctuations that a standard GP might exhibit.

In Figure 7.2, where a strictly increasing sinusoidal input was used, the monotonic GP clearly adhered to the expected functional form. It avoided the oscillatory deviations seen in the standard GP and provided sharper confidence intervals. This matched theoretical predictions — since the sinusoid is strictly increasing over the domain shown, the monotonic GP was not constrained inappropriately but instead guided to fit better.

10.2 | Handling Step Changes and Sharp Transitions

The performance of monotonic GPs in the presence of sharp functional transitions, as tested in Figures 7.3 and 7.4, revealed key strengths and limitations. Monotonic GPs managed to track the general trend while respecting the shape constraint, whereas the standard GP often exhibited erratic behavior like overshooting before or after the step, and displaying large uncertainty bands. However, monotonic GPs naturally tend to smooth over step transitions because the derivative constraints do not directly encode the existence of discontinuities. This is consistent with the theory: GPs, even when monotonic, remain smooth function models and thus handle steps by steep slopes rather than discontinuities. Nonetheless, the confidence intervals in the monotonic model remained more calibrated and tight, as shown in Figure 7.3, indicating that even under rapid transitions, monotonic constraints help avoid implausible non-monotonic spikes and dips that the standard GP is prone to.

10.3 | Effect of Noise and Data Sparsity

In Figure 7.1, we observe the impact of reduced observational noise. The standard GP, with fewer data points and less noise, became overconfident in regions lacking information, sometimes forming non-monotonic fits that deviated from the true function. In contrast, the monotonic GP retained its shape constraint, producing a smoother fit and appropriately reflecting uncertainty where data was limited. This behavior underscores the role of monotonic priors as a form of regularization by injecting inductive bias that stabilizes learning in the face of limited or low-noise data.

10.4 | Summary Performance

The plot (Figure 7.5) visually consolidates these observations. The monotonic GP consistently aligns more closely with the true underlying function, with confidence intervals that are narrower and more informative. The standard GP, while more flexible, frequently diverges from the ground truth and is more sensitive to noise and sharp transitions. These results validate the practical utility of monotonic GPs in scenarios where the target function is known to follow a monotonic trend.

11 | Conclusion

The results presented in this study affirm the theoretical benefits of incorporating monotonicity constraints into Gaussian Process models, particularly using the framework of virtual derivative observations introduced by Riihimäki and Vehtari[5]. By enforcing monotonicity through probabilistic derivative signs and applying Expectation Propagation for approximate inference, the model achieves more reliable predictions in cases where the true function adheres to a known trend. The monotonic GP demonstrates improved data efficiency, reduced uncertainty, and better generalization in low-noise or low-data scenarios.

Importantly, this approach has promising applications in scientific fields such as paleoclimatology, where age-depth relationships in sediment cores

are inherently monotonic. In such domains, incorporating domain knowledge into the GP model ensures that the reconstructed chronology does not violate physical constraints. The ability to encode monotonicity directly through the likelihood enables more interpretable and physically plausible modeling of age-depth profiles, enhancing the reliability of temporal reconstructions from proxy records.
However, the findings also highlight that such constraints must be used with caution. When applied to inherently non-monotonic functions, the monotonic GP exhibits bias and fails to capture essential dynamics. Thus, prior knowledge must be leveraged judiciously.
Overall, monotonic GPs provide a powerful and interpretable tool for function approximation, especially when domain knowledge supports the use of structural priors. Their ability to reduce model variance while maintaining theoretical rigor makes them well-suited for scientific and real-world applications where monotonicity is known or strongly suspected.

12 | Acknowledgement

I would like to express my deepest gratitude to my principal investigator, Dr. Bedartha Goswami, for his invaluable guidance, encouragement, and mentorship throughout the course of this project.

I would also like to sincerely thank Riihimäki and Vehtari for their pioneering work on monotonic Gaussian processes, which laid the theoretical foundation for this report. Their paper was both an inspiration and a cornerstone reference for my implementation and understanding.

Finally, I extend my heartfelt thanks to all those who supported me along the way — whether through technical discussions, emotional encouragement, or simply believing in my work. Your support made this journey possible and meaningful.

13 | References

[1] Python code.

[2] Stella Birlo, Wojciech Tylmann, and Bernd Zolitschka. Bayesian age–depth modelling applied to varve and radiometric dating to optimize the transfer of an existing high-resolution chronology to a new composite sediment profile from holzmaar (west eifel volcanic field, germany). *Geochronology*, 5(1):65–90, 2023.

[3] S. F. M. Breitenbach, K. Rehfeld, B. Goswami, J. U. L. Baldini, H. E. Ridley, D. J. Kennett, K. M. Prufer, V. V. Aquino, Y. Asmerom, V. J. Polyak, H. Cheng, J. Kurths, and N. Marwan. Constructing proxy records from age models (copra). *Climate of the Past*, 8:1765–1779, 2012.

[4] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[5] J. Riihimäki and A. Vehtari. Gaussian processes with monotonicity information. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.