Advanced Methods in Applied Statistics

Paper Summary: AutoIP: A United Framework to Integrate Physics into Gaussian Processes

Alexandra Haslund-Gourley and Søren Jefsen

March 5, 2025

1 Introduction

When fitting a model to physical data, one must balance myriad constraints: minimizing the misfit between the data and the proposed function, requiring few training data points, ensuring the function remains physically realistic, and quantifying the model's inherent uncertainty. The authors of *AutoIP: A Unified Framework to Integrate Physics into Gaussian Processes* Long et al. (2022) introduce a method called Automatically Incorporating Physics (AutoIP), which aims to address all of these challenges. Their approach leverages Gaussian Processes (GPs) to predict functions that fit training data while enforcing physically motivated partial differential equations (PDEs) as constraints. This integration of physics into GPs enhances both prediction accuracy and uncertainty quantification.

In this report, we first provide the necessary background on GPs before outlining the steps of AutoIP. We then summarize the experiments and results of the authors, followed by our commentary and suggestions for future directions.

1.1 Gaussian Processes

A Gaussian Process (GP) defines a probability distribution over possible functions that interpolate a given set of data points. As shown in Equation 1, GPs are parameterized by a mean function $\mu(x)$, which defines the expected function value at each point x, and a covariance function (kernel) $K(x, x')$, which encodes correlations between function values at different points, governing properties like smoothness and periodicity.

$$f(x) \sim \mathcal{GP}(\mu(x), K(x, x')) \tag{1}$$

Given training data (X, y), GP regression predicts function values y^* at unseen points X^* by first modeling a joint Gaussian distribution over both observed and unseen points, and then conditioning on the observed data. This yields the following expressions for the predictive mean (μ^*) in Equation 2 and variance (σ^{*2}) in Equation 3. The full derivation of these results is included in the appendix.

$$\mu^* = K(X^*, X)K(X, X)^{-1}y. \tag{2}$$

$$\sigma^{*2} = K(X^*, X^*) - K(X^*, X)K(X, X)^{-1}K(X, X^*). \tag{3}$$

1.2 AutoIP

The AutoIP framework begins by selecting a physical system, obtaining N training observations $D = \{(z_1, y_1), ..., (z_N, y_N)\}$, and defining a physically motivated PDE that fully or partially describes the system. As an illustrative example, the authors use a variant of the Allen–Cahn reaction-diffusion PDE in Equation 4, which relates input values $z_i = (x_i, t_i)$ to diffusion values y_i. The equation models the target function $u(x, t)$, an unknown source term $g(x, t)$, and learnable parameters v and γ.

$$\partial_t u - v\partial_x^2 u + \gamma u(u^2 - 1) + g(x, t) = 0. \tag{4}$$

Next, they sample a set of collocation points X^* from the function domain where the PDE constraints will be enforced. Using these points and the training data, they construct a covariance function that encapsulates both the standard GP covariance terms and the covariance of PDE-derived function derivatives. These derivatives are obtained using the kernel differentiation trick shown in Equation 5 for two differential operators $\hat{A}$ and $\hat{B}$:

$$\text{cov}(\hat{A}u(z_1), \hat{B}u(z_2)) = \hat{A}\hat{B}\kappa(z_1, z_2). \tag{5}$$

To integrate these additional derivative terms into a GP, the data and collocation points are combined into an input vector f, and the corresponding covariance matrices are arranged into a block-diagonal matrix Σ. For simplicity, the mean function ($\mu(x)$) is set to 0. This results in a joint Gaussian prior over all components of the PDE, as shown in Equation 6:

$$p(f) = \mathcal{N}(f|0, \Sigma). \tag{6}$$

This prior is then evaluated with two likelihoods: A likelihood to fit the function to observed data 7 and virtual likelihood ensuring that sampled functions satisfy the PDE 8.

$$p(y|f) = \mathcal{N}(y|u, \beta^{-1}I), \tag{7}$$

$$p(0|f) = \mathcal{N}(0|u_{bt} - vu_{bxx} + \gamma u_b \circ (u_b \circ u_b - 1) + g, vI). \tag{8}$$

Multiplying these likelihoods with the GP prior results in the joint probability distribution 9.

$$p(f, y, 0) = \mathcal{N}(f|0, \Sigma)\mathcal{N}(y|u, \beta^{-1}I)\mathcal{N}(0|\text{PDE}, vI). \tag{9}$$

The goal is to compute the posterior distribution $p(y^*|f, y, \text{PDE})$, which normally requires marginalizing out f. However, the complexity of the virtual likelihood makes this posterior intractable. To bypass this issue, the authors use a variational evidence lower bound (ELBO) approximation to infer the distribution.

1.3 Remarks

The authors assess AutoIP by first comparing its performance to standard GP regression. They then evaluate its robustness under different conditions, including varying the training data size, introducing noise, applying different PDEs, using incomplete (AutoIP-I) and complete (AutoIP-C) differential equations, and testing across various physical systems. Model performance is assessed using root mean square error (RMSE) and mean negative log-likelihood (MNLL) between the predicted and ground-truth functions.

We illustrate their conclusions using a simple pendulum example, shown in Figure 1. We can see that standard GPs fit well within the training regime but fail to extrapolate, simply reverting to the prior distribution when making predictions outside the training region. Extrapolation is improved from the standard GP by encoding an incomplete PDE (AutoIP-I), but even further enhanced for complete PDEs (AutoIP-C).

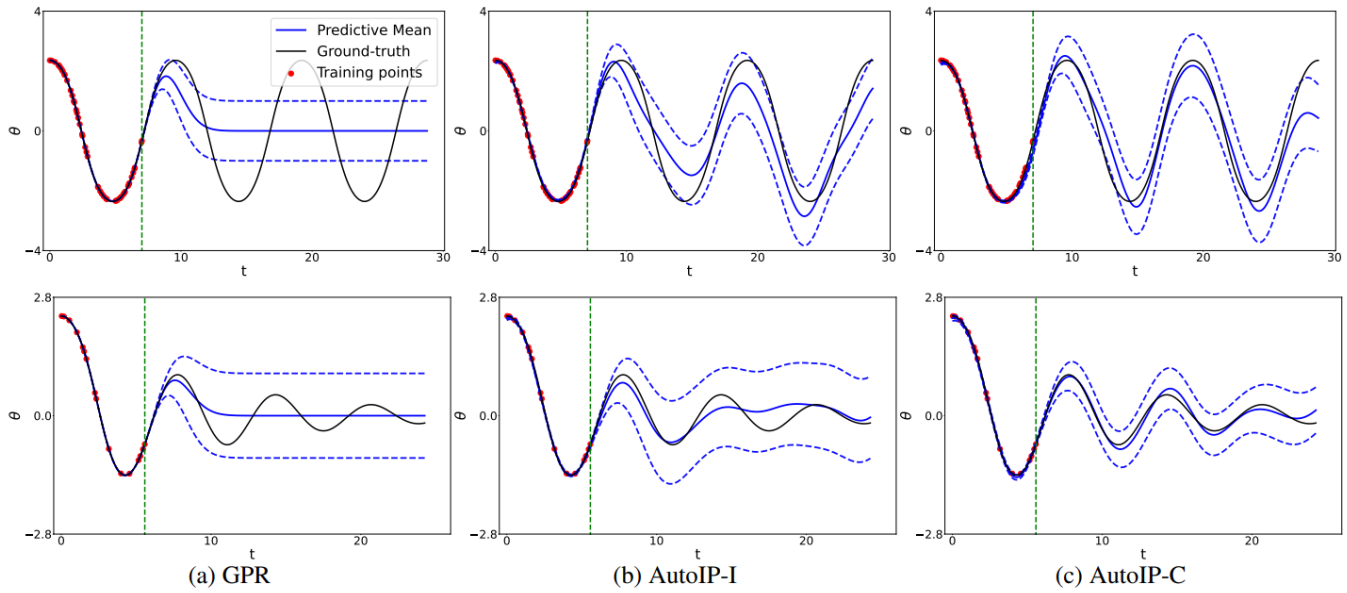
They find that AutoIP demonstrates robustness when trained on sparse or noisy data, likely because it can leverage the physics informed priors to maintain stability. In addition, incorporating additional collocation points from the domain improves accuracy by refining derivative estimates and improving predictive performance.

We find these results quite exciting but are also aware of the computational overhead of this method. Since the general GP method requires computing the inverse of the covariance matrix, which scales cubically with the number of data points, AutoIP's added complexity from enforcing PDE constraints could further increase computational costs, making scalability a potential challenge. This is particularly relevant because more complex and descriptive PDEs, as well as an increased number of collocation points, improve model performance. As a result, we are left wondering about the types of problems for which this method is best suited, especially when considering the balance between computational expense and predictive accuracy.

Furthermore, we are curious whether this method could be extended to handle non-stationary PDEs, in which the nature of the terms in the equation evolves over time. This could provide a more dynamic model, capturing the time-dependent shifts in physical systems. The ability to incorporate time-varying constraints into the GP framework could unlock new applications, allowing for more nuanced modeling of complex systems where the governing physics change over time.

2 Conclusion

AutoIP offers a principled approach to incorporating physical knowledge into Gaussian Processes, enabling accurate predictions, robust extrapolation from input data, and effective uncertainty quantification, all while ensuring compliance with physical constraints. The flexibility of the AutoIP framework allows for the integration of a wide range of PDEs, making it widely applicable across various scientific fields. Future research could explore extending AutoIP to address more complex physical systems, including non-stationary PDEs that evolve over time.



(a) GPR (b) AutoIP-I (c) AutoIP-C

Figure 1: Prediction in a nonlinear pendulum system with exact training examples. First row shows results without damping. Second row shows results with damping. Dashed lines are predictive mean ± standard deviation. Vertical line is the boundary of the training region.
Here, GPR refers to Standard Gaussian Processes, AutoIP-I means Incomplete Differential Equations and AutoIP-C indicates Complete Differential Equations.

A Gaussian Processes - Extended Explanation

A Gaussian process (GP) is fully specified by a mean function $\mu(x)$ and a covariance function (or kernel) $K(x, x')$ as shown in Equation 10.

$$f(x) \sim \mathcal{GP}(\mu(x), K(x, x')). \tag{10}$$

Here, $\mu(x)$ defines the expected value of the function at each point x, often assumed to be zero for simplicity ($\mu(x) = 0$). The kernel function $K(x, x')$ encodes the covariance between function values at different points, governing properties like smoothness and periodicity. A common choice for $K(x, x')$ is the Radial Basis Function (RBF) kernel shown in Equation 11, which enforces smoothness, as closer points will have higher covariance values. This structure enables the GP to generate continuous, plausible function approximations.

$$K(x, x') = \sigma_{RBF}^2 e^{-\frac{1}{2l^2}(x-x')^2}. \tag{11}$$

Putting this all together, for a set of training inputs $X = \{x_1, x_2, ..., x_n\}$, the function values $\mathbf{f}$ follow:

$$\mathbf{f}|X \sim \mathcal{N}(\mu(X), k(X, X)). \tag{12}$$

A.0.1 GP Regression

We can now use this formulation of the joint multivariate distribution over the points we are given (X, y) to make inferences about new points (X^*, y^*). In the regression setting, we assume there is noise in our observations, which we model by adding a random noise term, $\boldsymbol{\epsilon}$, drawn from a normal distribution centered at zero and parameterized by σ_n^2. This is shown in Equation 13:

$$\mathbf{y} = \mathbf{f} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_n^2 I). \tag{13}$$

Thus, the likelihood of the observations given the function values is:

$$p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma_n^2 I). \tag{14}$$

Given this, and using the prior distribution $p(\mathbf{f}|X)$, we can condition $\mathbf{y}$ on the training inputs X, leading to the following distribution for $\mathbf{y}$:

$$\mathbf{y}|X \sim \mathcal{N}(\mu(X), K(X,X) + \sigma_n^2 I). \tag{15}$$

Joint Distribution

For both the observed points $\mathbf{y}$ and the new test points $\mathbf{y}^*$, we can express the joint distribution of the function values at these points. For X as training inputs and X^* as test inputs, we have:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(X) \\ \mu(X^*) \end{bmatrix}, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X^*) \\ K(X^*,X) & K(X^*,X^*) \end{bmatrix} \right). \tag{16}$$

We can simplify the joint distribution to:

$$p(\mathbf{y}, \mathbf{y}^*) = \mathcal{N}\left(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \right). \tag{17}$$

This form allows us to marginalize over the observed values $\mathbf{y}$, leaving us with a Gaussian distribution for the predictions at the new test points:

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{a}, A).$$

Conditional Distribution and Prediction

The conditional distribution for the test points $\mathbf{y}^*$ given the observations $\mathbf{y}$ is then derived using Bayes' rule, and by the conditional formula for Gaussian Derived in Rasmussen and Williams (2006).

$$p(\mathbf{y}^*|\mathbf{y}) = \frac{p(\mathbf{y}^*, \mathbf{y})}{p(\mathbf{y})} = \mathcal{N}\left(\mathbf{a} + BC^{-1}(\mathbf{y} - \mathbf{b}), A - BC^{-1}B^\top\right).$$

Substituting the kernel values, we obtain the following expressions for the mean and variance of the predictions at the new inputs:

$$\mathbb{E}[\mathbf{y}^*] = \mu(X^*) + K(X^*,X)[K(X,X) + \sigma_n^2 I]^{-1}(\mathbf{y} - \mu(X)).$$

If $\mu(X) = 0$, this simplifies to:

$$\mathbb{E}[\mathbf{y}^*] = K(X^*,X)[K(X,X) + \sigma_n^2 I]^{-1}\mathbf{y}.$$

The variance of the predictions is:

$$\mathrm{Var}[\mathbf{y}^*] = K(X^*,X^*) - K(X^*,X)[K(X,X) + \sigma_n^2 I]^{-1}K(X,X^*), \tag{18}$$

Thus, we now have a closed-form solution to make predictions for new data points. The computational challenge here lies in calculating $[K(X,X) + \sigma_n^2 I]^{-1}$, which can become computationally expensive, particularly for large datasets.

Hyperparameter Optimization

The final step in GP regression is the optimization of the kernel's hyperparameters, such as σ_{RBF}^2 and l, which control the scale and length scale of the RBF kernel. These parameters can be optimized by maximizing the log marginal likelihood, which is given by:

$$\mathbf{argmax}_{l,\sigma_{RBF}^2} \log p(\mathbf{y}|X). \tag{19}$$

This step ensures that the GP is best suited to model the data, providing accurate predictions and uncertainty quantification .

References

Long, D., Wang, Z., Krishnapriyan, A., Kirby, R., Zhe, S., and Mahoney, M. (2022). Autoip: A united framework to integrate physics into gaussian processes.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.