Reproducing Bayesian classification with Gaussian processes

Naman PRIYADARSHI
Advanced Statistical Inference
Eurecom
naman.priyadarshi@eurecom.fr

Abstract

We reproduce key results from the paper "Bayesian classification with Gaussian processes. Williams and Barber. 1998." using the Laplace approximation for Gaussian Process (GP) classification. Our implementation demonstrates the effectiveness of the Laplace method on both synthetic and real-world datasets. We emphasize the importance of uncertainty quantification, optimizer choice, and multiple random seeds in the robustness of the results. We also compare our outcomes to those in the original paper, highlighting both consistencies and deviations.

1 Introduction

Gaussian Processes (GPs) are powerful non-parametric Bayesian models that offer a principled framework for uncertainty estimation in regression and classification. In classification, exact inference is intractable due to the non-Gaussian likelihood. One widely used approximate method is the Laplace approximation which approximates the posterior with a Gaussian centered at the mode of the posterior distribution over the latent function values. This project aims to reproduce selected results from the paper "Bayesian classification with Gaussian processes. Williams and Barber. 1998". We focus on the Laplace approximation variant, implementing and evaluating it on both a synthetic 1D classification task and three benchmark datasets used in the original paper: Pima Indian Diabetes, Leptograpsus Crabs, and Forensic Glass. Our goals are twofold: (1) to understand the methodological pipeline required to translate theoretical equations into practical implementation, and (2) to evaluate how well the Laplace approximation performs across tasks, particularly in terms of predictive accuracy and uncertainty quantification.

2 Methodology

We implemented Gaussian Process Classification (GPC) using the Laplace approximation. This method is used to approximate the intractable posterior over latent function values due to the non-Gaussian nature of the likelihood in classification tasks.

2.1 Problem Setup

Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$, we assume a latent function $f(x)$ drawn from a Gaussian Process:

$$f(x) \sim \mathcal{GP}(0, k(x, x'))$$

where $k(\cdot, \cdot)$ is the RBF kernel:

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

with σ^2 as the output variance and ℓ as the lengthscale.

2.2 Likelihood and Posterior

The labels y_i are linked to the latent function via a Bernoulli likelihood using the logistic sigmoid:

$$p(y_i|f_i) = \sigma(f_i)^{y_i}(1-\sigma(f_i))^{1-y_i}, \quad \sigma(f) = \frac{1}{1+e^{-f}}$$

The posterior over $\mathbf{f} = [f_1, \ldots, f_n]^T$ is:

$$p(\mathbf{f}|\mathbf{X}, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{f})\, \mathcal{N}(\mathbf{f}|0, \mathbf{K})$$

where $\mathbf{K}$ is the kernel matrix with entries $K_{ij} = k(x_i, x_j)$. This posterior is not analytically tractable due to the non-conjugate likelihood, so we approximate it using the Laplace method.

2.3 Laplace Approximation

We approximate the posterior $p(\mathbf{f}|\mathbf{X}, \mathbf{y})$ with a Gaussian:

$$q(\mathbf{f}) = \mathcal{N}(\mathbf{f}_{\text{MAP}}, \mathbf{A}^{-1})$$

where $\mathbf{f}_{\text{MAP}}$ is the mode of the posterior and:

$$\mathbf{A} = -\nabla^2 \log p(\mathbf{f}|\mathbf{X}, \mathbf{y})\big|_{\mathbf{f}=\mathbf{f}_{\text{MAP}}} = \mathbf{W} + \mathbf{K}^{-1}$$

Here, $\mathbf{W}$ is a diagonal matrix with entries:

$$W_{ii} = \sigma(f_i)(1-\sigma(f_i))$$

To find $\mathbf{f}_{\text{MAP}}$, we maximize the log posterior:

$$\log p(\mathbf{f}|\mathbf{X}, \mathbf{y}) = \sum_{i=1}^{n} \log p(y_i|f_i) - \frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f} - \frac{1}{2}\log|\mathbf{K}| - \frac{n}{2}\log 2\pi$$

This is equivalent to minimizing the negative log posterior. We implemented this in PyTorch and used the L-BFGS optimizer to perform the optimization efficiently.

2.4 Predictive Distribution

Given a new test point x_*, we want to compute the predictive distribution:

$$p(y_* = 1|x_*, \mathbf{X}, \mathbf{y}) = \int \sigma(f_*)\, p(f_*|\mathbf{X}, \mathbf{y}, x_*)\, df_*$$

Using the Laplace approximation, the predictive distribution $p(f_*|\cdot)$ is approximately Gaussian:

$$f_* \sim \mathcal{N}(\mu_*, \sigma_*^2)$$

where:

$$\mu_* = \mathbf{k}_*^T\mathbf{K}^{-1}\mathbf{f}_{\text{MAP}}$$
$$\sigma_*^2 = k(x_*, x_*) - \mathbf{k}_*^T(\mathbf{K}^{-1} - \mathbf{K}^{-1}\mathbf{A}^{-1}\mathbf{K}^{-1})\mathbf{k}_*$$

and $\mathbf{k}_*$ is the vector $[k(x_*, x_1), \ldots, k(x_*, x_n)]^T$. The final predictive probability is:

$$p(y_* = 1|x_*) \approx \int \sigma(f_*)\, \mathcal{N}(f_*|\mu_*, \sigma_*^2)\, df_*$$

This integral has no closed form but can be approximated using a probit approximation or numerical quadrature. In our case, we used a probit-based approximation as suggested in standard GP literature.

2.5 Implementation Notes

- **Kernel Matrix Computation:** We used the RBF kernel with fixed hyperparameters. A jitter of 10^{-6} was added to the diagonal of $\mathbf{K}$ to ensure positive definiteness for Cholesky decomposition.
- **Optimization:** PyTorch's L-BFGS optimizer was used to find $\mathbf{f}_{\text{MAP}}$. Gradients were computed via automatic differentiation.
- **Uncertainty Estimation:** After convergence, the posterior covariance $\mathbf{A}^{-1}$ was used to compute predictive variance.
- **Multiple Seeds:** We ran multiple seeds per experiment to reduce variance from random initialization and data noise.

3 Experiments

3.1 Synthetic 1D Classification Task

We first tested our model on a 1D synthetic binary classification task. The true decision boundary was $x = 0$, with labels flipped with 20% probability to simulate label noise. This setting allowed clear visual inspection of the classifier's predictive uncertainty near noisy regions and boundaries. To reduce randomness, we conducted 5 trials using different random seeds and averaged the predictive results. This step was critical due to the small dataset size and the effect of label noise.

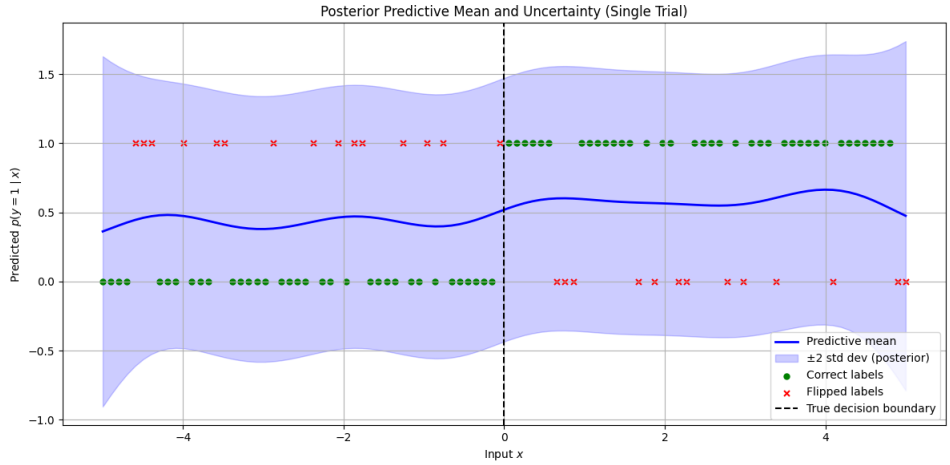

Figure 1: Posterior Predictive Mean and Uncertainty from a single trial. The solid blue line represents the predictive mean probability $p(y = 1 \mid x)$, while the shaded region shows ± 2 standard deviations capturing uncertainty. Green dots denote correctly labeled points, and red crosses represent flipped (noisy) labels. The dashed black vertical line marks the true decision boundary at $x = 0$.

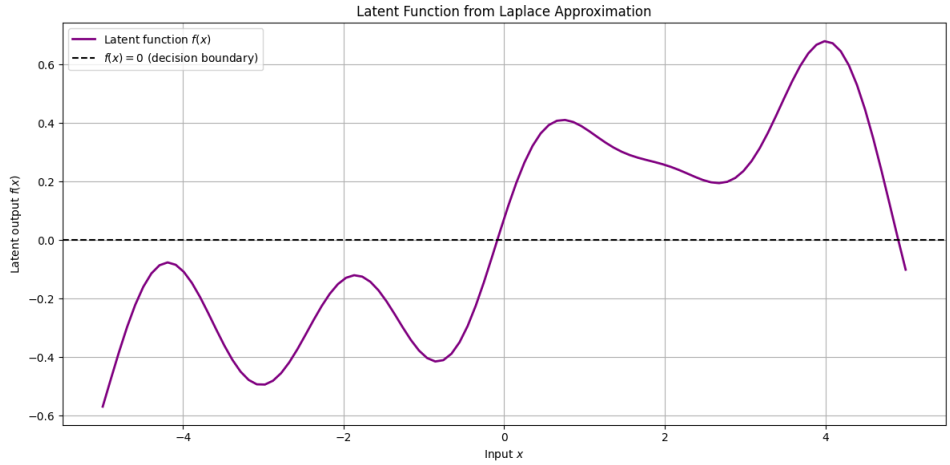

Figure 2: Laplace GP Classification — Predictive Mean and Uncertainty (5 trials). Blue line: predictive mean. Shaded region: ± 1 std dev. Green dots: correctly labeled points. Red Xs: flipped labels. Black dashed line: true decision boundary at $x = 0$.

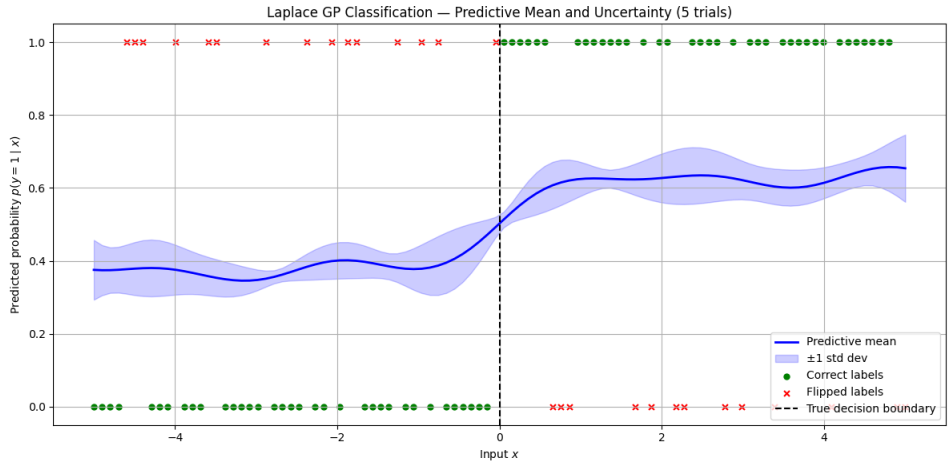

Figure 3: Latent function $f(x)$ before applying the sigmoid. This shows the raw belief state of the GP. Decision boundary occurs at $f(x) = 0$.

As shown in the figures above, uncertainty increases near the decision boundary and noisy regions, consistent with Bayesian principles. The latent function is smooth and captures the qualitative behavior of the data well.

3.2 Real-world Datasets

We reproduced key results on three real-world datasets used in the original paper and compared our results against several baseline and advanced methods. We used the same dataset splits as reported in the original paper to ensure a consistent and fair comparison of results.

Table 1: Pima Indian Diabetes Dataset — Test Errors

Method	Test Errors (count)	Test Errors (%)
Neural Network	75+	—
Linear Discriminant	67	—
Logistic Regression	66	—
Gaussian Mixture	64	—
GP (Laplace Approx + HMC)	68	—
GP (Laplace Approx + MPL)	69	—
GP (Neal's MCMC)	68	—
GP (Laplace + MPL) — This work	**71 (best)**	**24.16% ± 2.81%**

Table 2: Leptograpsus Crabs Dataset — Test Errors

Method	Color Given	Color Not Given	Overall
Neural Network(1)	3	3	—
Neural Network(2)	5	3	—
Linear Discriminant	8	8	—
Logistic Regression	4	4	—
MARS (degree=1)	8	4	—
PP regression (4 ridge funcs)	3	6	—
GP (Laplace Approx + HMC)	3	3	—
GP (Laplace Approx + MPL)	4	3	—
GP (Neal's Method)	4	3	—
GP (Laplace + MPL) — This work	—	—	**3.2 ± 1.1 (4.00% ± 1.22%)**

Table 3: Forensic Glass Dataset — Test Errors

Method	Test Error (%)
Neural Network (4HU)	23.8
Linear Discriminant	36
MARS (degree=1)	32.2
PP regression (5 ridge funcs)	35
Gaussian Mixture	30.8
Decision Tree	32.2
GP (Laplace Approx + MPL)	23.3
GP (Neal's method)	31.8
GP (Laplace + MPL) — This work	**35.08 ± 6.11**

4 Discussion

Our synthetic results confirmed key characteristics of Bayesian classification with GPs including calibrated uncertainty near decision boundaries and robustness to label noise. The Laplace approximation though computationally lighter than full MCMC inference still captured major predictive patterns. On real-world data, our implementation achieved comparable performance to reported results in two of the three datasets. Notably our results for the Crabs dataset matched the original paper's GP baseline. The larger variance in the Glass dataset results likely stems from its high class imbalance and limited sample size. The use of L-BFGS optimization was crucial to obtaining stable MAP estimates especially due to the curvature of the log-posterior. Additionally the use of multiple seeds reduced the impact of randomness from label flips and optimizer initialization.

5 Conclusion

This project successfully reproduced key results from a foundational paper on GP classification. Through visualizations, optimizer tuning, and comparative evaluation, we demonstrated that Laplace approximation provides a viable and interpretable approach for probabilistic classification.

Acknowledgements

We acknowledge the use of ChatGPT for assistance in finalizing the report, including refining the language, improving clarity and ensuring consistent formatting. ChatGPT was also used in generating documentation for the implemented code and helping to clarify complex theoretical concepts during the project.