# Pseudo-Marginal Methods for Gaussian Process Classification

February 2025

## 1 Gaussian Process Classification and MoM-PM Algorithm

We evaluate the performance of the median-of-means (MoM) Pseudo-Marginal (PM) method in the context of Gaussian Process (GP) classification. This section presents the experimental setup, Bayesian inference procedure, and results comparing MoM-PM against standard Importance Sampling (IS) in the Pseudo-Marginal Metropolis-Hastings (PM-MH) framework.

### 1.1 Gaussian Process Classifier

Let

$$X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$$

be a set of $n$ input vectors with associated binary labels

$$\mathbf{y} = (y_1, \ldots, y_n)^\top, \quad y_i \in \{-1, 1\}.$$

The latent function values $\mathbf{f} = (f_1, \ldots, f_n)^\top$ follow a zero-mean Gaussian Process:

$$\mathbf{f} \sim \mathcal{N}(\mathbf{0}, K(X, \theta)),$$

where $K(X, \theta)$ is the covariance matrix determined by a kernel function $k(x_i, x_j \mid \theta)$:

$$k(x_i, x_j \mid \theta) = \sigma^2 \exp\left(-\frac{1}{2}(x_i - x_j)^\top A (x_i - x_j)\right),$$

with

$$A^{-1} = \mathrm{diag}(\tau_1^2, \ldots, \tau_d^2), \quad \theta = [\tau_1, \ldots, \tau_d, \sigma].$$

The likelihood function for classification follows a probit model:

$$p(y_i \mid f_i) = \Phi(y_i f_i),$$

where $\Phi$ denotes the standard normal cumulative distribution function.

In this experiment, we adopt the following prior distributions for the kernel parameters:

$$p(\tau_i) = \mathcal{G}(1, 1/\sqrt{d}), \quad p(\sigma) = \mathcal{G}(1.2, 0.2).$$

We generate synthetic data from a Gaussian Process (GP) with true parameters $\tau_i = 0.35$ and $\sigma = 2.08$, using $n = 50$ and $d = 2$. All experimental settings follow the setup in [1].

## 1.2 Bayesian Inference via Pseudo-Marginal Methods

Our goal is sampling from the posterior distribution of the kernel parameters:

$$p(\theta \mid X, \mathbf{y}) \propto p(\theta) p(\mathbf{y} \mid X, \theta).$$

Since the marginal likelihood

$$p(\mathbf{y} \mid X, \theta) = \int p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid X, \theta) d\mathbf{f}$$

is intractable, we employ a Pseudo-Marginal Metropolis-Hastings (PM-MH) algorithm using an importance-sampling (IS) estimator.

### 1.2.1 Importance Sampling Estimator

The unbiased IS estimator for $p(\mathbf{y} \mid X, \theta)$ is given by

$$\hat{p}(\mathbf{y} \mid \theta) = \frac{1}{N_{\text{IS}}} \sum_{i=1}^{N_{\text{IS}}} \frac{p(\mathbf{y} \mid \mathbf{f}_i) p(\mathbf{f}_i \mid X, \theta)}{q(\mathbf{f}_i)},$$

where:

- $\mathbf{f}_i$ are sampled from a proposal distribution $q(\mathbf{f})$.
- $N_{\text{IS}}$ is the number of importance samples.

We use a Laplace approximation to define $q(\mathbf{f})$. The number of particles used is 64.

## 1.3 Experimental Setup

To evaluate the effectiveness of the MoM-based pseudo-marginal method, we compare the following four approaches:

- **PM-IS:** Standard Pseudo-Marginal Metropolis-Hastings using an Importance Sampling (IS) estimator.
- **MoM-biased:** Median-of-Means (MoM) applied to IS without outer-loop correction.

- **MoM:** MoM applied to IS, with an additional outer-loop importance correction.
- **Double-MoM:** MoM applied both in the inner IS estimator and in the outer-loop correction.

In the MoM-based pseudo-marginal methods, the standard IS estimator $\hat{p}(\mathbf{y} \mid \theta)$ is replaced by the MoM estimator $\hat{p}_{\text{MoM}}(\mathbf{y} \mid \theta)$, using $K = 4$ groups. Since the MoM estimator introduces bias, the resulting Markov chain targets a perturbed posterior distribution. To correct this bias, we apply an outer-loop importance correction in the MoM and Double-MoM variants, making the estimator unbiased.

*Remark* 1. In our approach, the inner loop refers to the Pseudo-Marginal Markov Chain (which estimates the likelihood via IS or MoM-IS), while the outer loop corresponds to the importance correction step, particularly in PM-MoM.

We run each method for 15,000 iterations, discarding the first 5,000 as burn-in. Gaussian random walk (GRW) proposals are applied to $\log \theta$ to ensure positivity.

The proposal variance is tuned to achieve an acceptance rate of approximately 7% in the PM-IS method, as suggested in [2]. In our experiments, PM-IS achieves an average acceptance rate of 6.79%, while PM-MoM reaches 21.24%.

## 1.4 Results and Analysis

Figure 1 shows the kernel-smoothed inefficiency factor distribution for 500 Markov Chains generated by PM-IS and PM-MoM.

Figure 2 presents the posterior distributions of $\sigma$ and $\tau$ under different methods.

We observe that:

- **MoM-biased** produces fewer outliers than PM-IS, suggesting improved mixing. A similar conclusion can be drawn from the inefficiency factor distribution in Figure 1.
- **MoM** effectively corrects the bias introduced by the MoM estimator, but it may lead to a lower effective sample size (ESS), resulting in heavier tails in the posterior distribution.
- **Double-MoM** mitigates the low ESS issue and improves robustness by applying MoM both in the inner IS estimator and the outer correction step.

Our results suggest that MoM-PM improves posterior robustness but may require further tuning to balance efficiency and mixing.

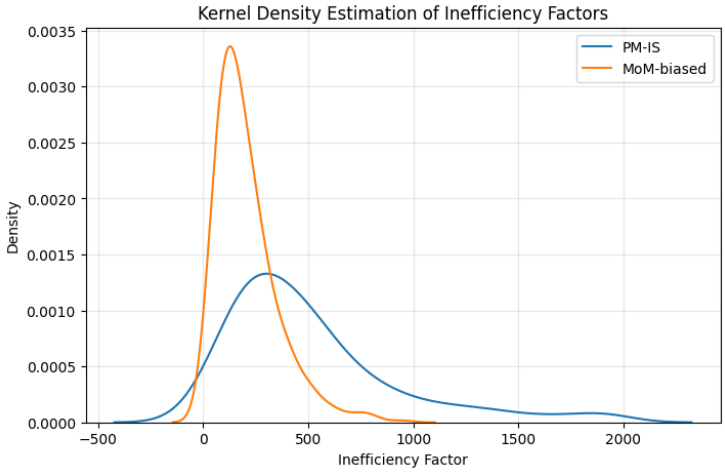



Figure 1: Kernel-smoothed inefficiency factor distribution of 500 Markov Chains generated by PM-IS and PM-MoM.

## consider adding

**[Consider adding: Quantitative comparison table of ESS, acceptance rate, and variance reduction.]**

**[Consider adding: Autocorrelation or trace plots to confirm mixing improvement.]**

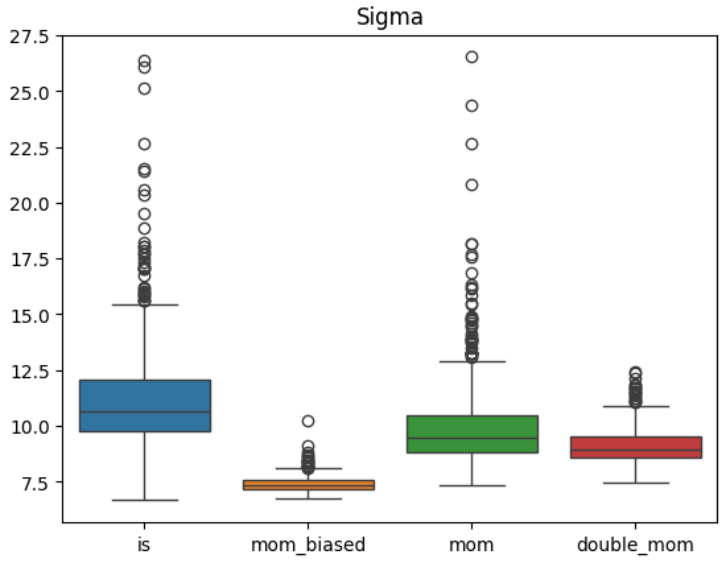
**[Consider adding: Discussion on when MoM-PM is most useful compared to PM-IS—e.g., small datasets, high noise settings.]**

**[Consider adding: Comparison of optimal number of particles determined by the noise of log-estimators.]**

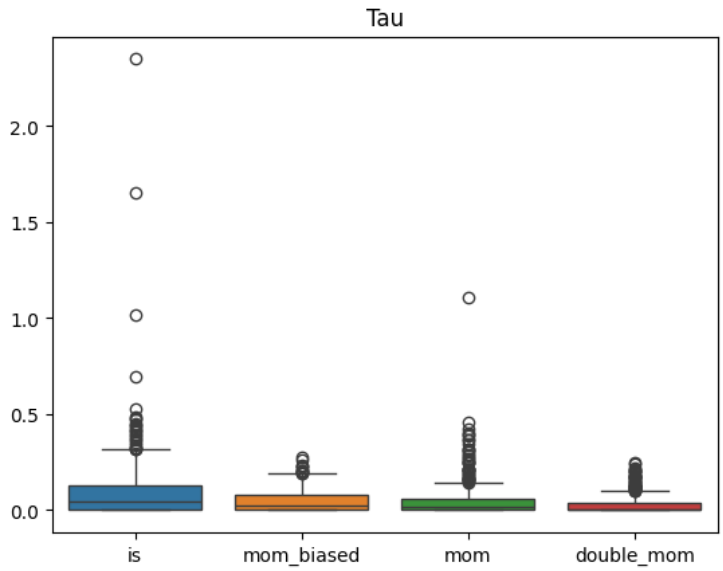
## References

[1] M. Filippone and M. Girolami, *"Pseudo-Marginal Bayesian Inference for Gaussian Processes,"* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014.

[2] C. Sherlock et al., *"On the Efficiency of Pseudo-Marginal Random Walk Metropolis Algorithms,"* The Annals of Statistics, 2015.




(a) Posterior of $\sigma$.




(b) Posterior of $\tau$.

Figure 2: Posterior distributions for kernel parameters under different methods.