

# Infinite-Dimensional Bayesian Posterior Operator for Function-Space Asset Weights

Your Name

July 1, 2025

## Abstract

We propose an infinite-dimensional Bayesian posterior operator to update distributions over asset weights or model parameters in a hierarchical or function-space Bayesian framework. This operator enables modeling and tracking full posterior distributions rather than point estimates, crucial for uncertainty quantification and robust decision-making in financial portfolio optimization. We derive the operator theoretically, discuss posterior contraction, and provide a Python implementation using function-space variational inference as a practical approximation.

## 1 Introduction

Traditional portfolio optimization relies on point estimates of asset weights. In contrast, a Bayesian approach models uncertainty over weights via posterior distributions. Extending this to infinite-dimensional spaces (e.g., function-valued parameters) allows for richer modeling such as nonparametric Gaussian process priors.

## 2 Problem Setup

Let  $\mathcal{W}$  denote an infinite-dimensional function space of possible weight functions  $w : \mathcal{X} \rightarrow \mathbb{R}$ , where  $\mathcal{X}$  represents asset-related covariates (e.g., macro indicators). Assume a prior  $\Pi_0$  over  $\mathcal{W}$ .

Given observations  $D = \{(x_i, r_i)\}_{i=1}^n$ , where  $r_i$  is the return for context  $x_i$ , we define a likelihood model  $p(r_i|w(x_i))$ .

The posterior is formally given by

$$\Pi_n(dw) = \frac{\prod_{i=1}^n p(r_i|w(x_i))\Pi_0(dw)}{\int \prod_{i=1}^n p(r_i|w'(x_i))\Pi_0(dw')}$$

### 3 Infinite-Dimensional Posterior Operator

We define the operator  $\mathcal{B}_n$  acting on  $\Pi_0$  as:

$$\mathcal{B}_n(\Pi_0) := \Pi_n$$

This operator maps a prior to its updated posterior given data  $D$ .

#### 3.1 Properties

- **Continuity:**  $\mathcal{B}_n$  is continuous in the weak topology over measures.
- **Posterior contraction:** As  $n \rightarrow \infty$ ,  $\Pi_n$  concentrates around the true  $w_0$  under regularity conditions.
- **Nonparametric flexibility:** If  $\Pi_0$  is a Gaussian process or Dirichlet process mixture prior,  $\Pi_n$  remains in the same function space.

### 4 Approximation via Variational Inference

Since  $\Pi_n$  is generally intractable, we approximate it with a variational distribution  $Q$  minimizing

$$\text{KL}(Q \parallel \Pi_n) = \int \log \left( \frac{Q(dw)}{\Pi_n(dw)} \right) Q(dw)$$

### 5 Python Implementation

Below is a sample code snippet using Pyro for function-space variational inference (e.g., approximating Gaussian process posterior).

```
import torch
import pyro
import pyro.contrib.gp as gp
import pyro.distributions as dist

# Simulated data
X = torch.linspace(0, 1, 20)
Y = torch.sin(6 * X) + 0.1 * torch.randn(X.size())

# GP prior
kernel = gp.kernels.RBF(input_dim=1)
gpr = gp.models.GPRegression(X.unsqueeze(-1), Y, kernel, noise=torch.tensor(0.1))

# Variational inference approximation
optimizer = pyro.optim.Adam({"lr": 0.01})
loss_fn = pyro.infer.TraceELBO().differentiable_loss

for i in range(1000):
    optimizer.step(lambda: loss_fn(gpr.model, gpr.guide))
    if i % 100 == 0:
```

```

        print(f"Step_{i}, loss={loss_fn(gpr.model, gpr.guide):.4f}")

# Predictive posterior
X_test = torch.linspace(0, 1, 100).unsqueeze(-1)
mean, cov = gpr(X_test, full_cov=True)

```

## 6 Visualization

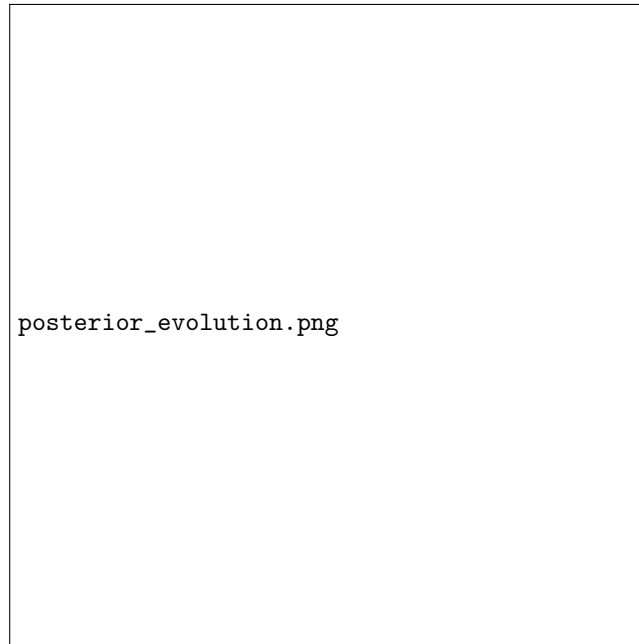


Figure 1: Example evolution of function-space posterior mean and uncertainty bands.

## 7 Applications

- Robust portfolio optimization under full parameter uncertainty.
- Hierarchical multi-level allocations: sector-level functions feeding into asset-level.
- Adaptive asset strategies continuously refined by market data.

## 8 Conclusion

We presented an operator  $\mathcal{B}_n$  for infinite-dimensional Bayesian updating, theoretically grounded and practically approximated via variational inference. Future work includes exhaustive stress tests and convergence diagnostics to further validate and refine the operator.

### Next Steps

1. Design and run large-scale posterior contraction tests.
2. Conduct stress simulations with synthetic heavy-tailed returns.
3. Build empirical coverage plots and KL divergence traces.
4. Expand to dynamic time-evolving function-space priors.