Mathematical Foundations of Uncertainty-Aware Forecasting: From Bayesian Inference to Sparse Variational Gaussian Processes

December 24, 2025

Abstract

This document provides a comprehensive mathematical treatment of the uncertainty quantification framework employed in our energy forecasting system. We begin with foundational concepts in Bayesian inference, progress through the theory of Gaussian processes, and culminate in a detailed exposition of sparse variational GP methods. Each section builds rigorously from first principles, deriving all key results with complete mathematical proofs. The treatment covers: (1) Bayesian inference and posterior computation, (2) Gaussian process priors and the function-space view, (3) exact GP regression and its computational limitations, (4) sparse approximations via inducing points, (5) variational inference and the evidence lower bound (ELBO), and (6) practical implementation considerations. This document serves as a self-contained mathematical reference for understanding every component of our uncertainty quantification pipeline.

Contents

1 Bayesian Inference: Foundations

1.1 The Bayesian Paradigm

The fundamental principle of Bayesian inference is the treatment of unknown quantities as random variables with probability distributions. This probabilistic framework enables principled uncertainty quantification.

Definition 1.1 (Bayes' Theorem)**.** Given a parameter $\theta \in \Theta$, observed data $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, and model likelihood $p(y|x, \theta)$, the posterior distribution over parameters is:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int_\Theta p(\mathcal{D}|\theta')p(\theta')d\theta'} \tag{1}$$

where:

- $p(\theta)$ is the *prior* distribution (beliefs before seeing data)
- $p(\mathcal{D}|\theta)$ is the *likelihood* (probability of data given parameters)
- $p(\mathcal{D})$ is the *marginal likelihood* or *evidence* (normalizing constant)
- $p(\theta|\mathcal{D})$ is the *posterior* distribution (beliefs after seeing data)

Remark 1.1. The evidence $p(\mathcal{D})$ is often intractable, requiring approximation methods such as Markov Chain Monte Carlo (MCMC), variational inference, or Laplace approximation.

1.2 Predictive Distribution

After observing training data $\mathcal{D}$, we make predictions at a new test point x_* by marginalizing over the posterior:

$$p(y_*|x_*, \mathcal{D}) = \int_\Theta p(y_*|x_*, \theta)p(\theta|\mathcal{D})d\theta \tag{2}$$

This *posterior predictive distribution* naturally incorporates two sources of uncertainty:

1. **Aleatoric uncertainty** (data noise): $\text{Var}[y_*|x_*, \theta]$
2. **Epistemic uncertainty** (parameter uncertainty): $\text{Var}_{\theta \sim p(\theta|\mathcal{D})}[\mathbb{E}[y_*|x_*, \theta]]$

1.3 Maximum A Posteriori (MAP) Estimation

When full posterior computation is intractable, a common approximation is the MAP estimate:

$$\theta_{\text{MAP}} = \arg\max_\theta p(\theta|\mathcal{D}) = \arg\max_\theta \left[\log p(\mathcal{D}|\theta) + \log p(\theta)\right] \tag{3}$$

Remark 1.2. MAP estimation provides a point estimate but loses uncertainty quantification—a key limitation that Gaussian processes address by maintaining full distributions over functions.

2 Gaussian Processes: The Function-Space View

2.1 From Weight-Space to Function-Space

Traditional parametric models (e.g., neural networks) place priors over weights:

$$p(\theta) \implies p(f) = p(f|\theta)p(\theta) \tag{4}$$

Gaussian processes *directly* specify a prior over functions:

$$f \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot)) \tag{5}$$

This function-space view provides several advantages:

- Direct uncertainty quantification over predictions
- Automatic calibration of complexity to data
- Principled extrapolation with increasing uncertainty
- Interpretable hyperparameters (lengthscales, amplitudes)

2.2 Definition and Properties

Definition 2.1 (Gaussian Process)**.** A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. A GP is completely specified by its mean function $m(\cdot)$ and covariance (kernel) function $k(\cdot, \cdot)$:

$$\begin{align} f(x) &\sim \mathcal{GP}(m(x), k(x, x')) \tag{6} \\ m(x) &= \mathbb{E}[f(x)] \tag{7} \\ k(x, x') &= \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))] \tag{8} \end{align}$$

For any finite set of points $X = \{x_1, \ldots, x_N\}$, the function values form a multivariate Gaussian:

$$\mathbf{f} = [f(x_1), \ldots, f(x_N)]^\top \sim \mathcal{N}(\mathbf{m}, K) \tag{9}$$

where:

$$\begin{align} \mathbf{m} &= [m(x_1), \ldots, m(x_N)]^\top \tag{10} \\ K_{ij} &= k(x_i, x_j) \tag{11} \end{align}$$

Theorem 2.1 (Consistency Property)**.** A GP satisfies the consistency property: if $\mathbf{f} \sim \mathcal{N}(\mathbf{m}, K)$ and $\mathbf{f}_A$ is any subset of $\mathbf{f}$, then:

$$p(\mathbf{f}_A) = \int p(\mathbf{f}) d\mathbf{f}_B \tag{12}$$

where $\mathbf{f} = [\mathbf{f}_A, \mathbf{f}_B]^\top$ is the partition into subsets A and B.

2.3 Kernel Functions

The kernel function $k(x, x')$ encodes our prior beliefs about the function's structure.

Definition 2.2 (Valid Kernel)**.** A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a valid kernel (positive definite) if for all finite sets $\{x_1, \ldots, x_N\} \subset \mathcal{X}$ and all $\mathbf{c} \in \mathbb{R}^N$:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j k(x_i, x_j) \geq 0 \tag{13}$$

Equivalently, the Gram matrix K with entries $K_{ij} = k(x_i, x_j)$ must be positive semi-definite.

2.3.1 Common Kernels

1. Radial Basis Function (RBF) / Squared Exponential Kernel:

$$k_{\text{RBF}}(x, x') = \sigma_f^2 \exp\left(-\frac{\|x - x'\|^2}{2\ell^2}\right) \tag{14}$$

Properties: Infinitely differentiable, stationary, universal approximator

Parameters: σ_f^2 (signal variance), ℓ (lengthscale)

2. Matérn Kernel:

$$k_{\text{Matérn}}(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{\ell}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu} r}{\ell}\right) \tag{15}$$

where $r = \|x - x'\|$, K_ν is the modified Bessel function, and ν controls smoothness.

Special cases:

- $\nu = 1/2$: Exponential kernel (continuous but not differentiable)
- $\nu = 3/2$: Once differentiable
- $\nu = 5/2$: Twice differentiable
- $\nu \to \infty$: Converges to RBF

3. Linear Kernel:

$$k_{\text{Linear}}(x, x') = \sigma_b^2 + (x - c)(x' - c) \tag{16}$$

Properties: Models linear trends, non-stationary

4. Periodic Kernel:

$$k_{\text{Periodic}}(x, x') = \sigma_f^2 \exp\left(-\frac{2\sin^2(\pi|x - x'|/p)}{\ell^2}\right) \tag{17}$$

Properties: Captures exact periodic patterns with period p

2.3.2 Kernel Composition

Kernels can be combined to create more expressive covariance structures:

Theorem 2.2 (Kernel Closure Properties). If k_1 and k_2 are valid kernels, then so are:

- Sum: $k(x,x') = k_1(x,x') + k_2(x,x')$
- Product: $k(x,x') = k_1(x,x') \cdot k_2(x,x')$
- Scaling: $k(x,x') = \alpha k_1(x,x')$ for $\alpha > 0$

Example 2.1 (Composite Kernel in Our System). We employ a sum of RBF and linear kernels:

$$k(x,x') = k_{\text{RBF}}(x,x') + k_{\text{Linear}}(x,x') \tag{18}$$

This captures both smooth non-linear patterns (RBF) and global trends (Linear).

3 Gaussian Process Regression

3.1 Model Specification

Consider the regression problem with training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ where:

$$f(x) \sim \mathcal{GP}(0, k(x,x')) \tag{19}$$

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_n^2) \tag{20}$$

The observations are noisy evaluations of the latent function f. Without loss of generality, we assume $m(x) = 0$ (can always subtract the mean).

3.2 Joint Distribution

For training inputs $X = \{x_1, \ldots, x_N\}$ and test inputs $X_* = \{x_{*1}, \ldots, x_{*M}\}$, the joint distribution of training and test function values is:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{21}$$

where:

$$K(X,X) \in \mathbb{R}^{N\times N}, \quad [K(X,X)]_{ij} = k(x_i, x_j) \tag{22}$$

$$K(X,X_*) \in \mathbb{R}^{N\times M}, \quad [K(X,X_*)]_{ij} = k(x_i, x_{*j}) \tag{23}$$

$$K(X_*,X_*) \in \mathbb{R}^{M\times M}, \quad [K(X_*,X_*)]_{ij} = k(x_{*i}, x_{*j}) \tag{24}$$

3.3 Posterior Predictive Distribution

Theorem 3.1 (GP Posterior). Given observations $\mathbf{y}$, the posterior distribution over test function values is:

$$\mathbf{f}_* | X_*, X, \mathbf{y} \sim \mathcal{N}(\bar{\mathbf{f}}_*, \text{Cov}(\mathbf{f}_*)) \tag{25}$$

where:

$$\bar{\mathbf{f}}_* = K(X_*,X)[K(X,X) + \sigma_n^2 I]^{-1}\mathbf{y} \tag{26}$$

$$\text{Cov}(\mathbf{f}_*) = K(X_*,X_*) - K(X_*,X)[K(X,X) + \sigma_n^2 I]^{-1}K(X,X_*) \tag{27}$$

Proof. This follows from standard Gaussian conditioning formulas. Let:

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix}\right) \tag{28}$$

Then the conditional distribution is:

$$\mathbf{a}|\mathbf{b} \sim \mathcal{N}(\bar{\mathbf{a}}, \bar{\Sigma}) \tag{29}$$
$$\bar{\mathbf{a}} = \boldsymbol{\mu}_a + \Sigma_{ab}\Sigma_{bb}^{-1}(\mathbf{b} - \boldsymbol{\mu}_b) \tag{30}$$
$$\bar{\Sigma} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba} \tag{31}$$

Applying this with $\mathbf{a} = \mathbf{f}_*$, $\mathbf{b} = \mathbf{y}$, $\boldsymbol{\mu}_a = \boldsymbol{\mu}_b = \mathbf{0}$, and the block covariance structure yields the result. □

3.4 Computational Complexity

The posterior mean and covariance require solving the linear system:

$$(K(X,X) + \sigma_n^2 I)\boldsymbol{\alpha} = \mathbf{y} \tag{32}$$

Computational costs:

- Matrix inversion/Cholesky decomposition: $O(N^3)$
- Storage: $O(N^2)$ for Gram matrix
- Prediction: $O(MN^2)$ for M test points

Computational Bottleneck

For large datasets ($N > 10{,}000$), exact GP inference becomes intractable. This motivates sparse approximations.

3.5 Hyperparameter Learning

GP hyperparameters $\boldsymbol{\theta} = \{\ell, \sigma_f^2, \sigma_n^2, \ldots\}$ are learned by maximizing the marginal likelihood (evidence):

$$\log p(\mathbf{y}|X, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^\top K_y^{-1}\mathbf{y} - \frac{1}{2}\log|K_y| - \frac{N}{2}\log(2\pi) \tag{33}$$

where $K_y = K(X,X) + \sigma_n^2 I$.

The three terms have interpretations:

1. $-\frac{1}{2}\mathbf{y}^\top K_y^{-1}\mathbf{y}$: Data fit (penalizes predictions far from observations)
2. $-\frac{1}{2}\log|K_y|$: Complexity penalty (penalizes overly complex models)
3. $-\frac{N}{2}\log(2\pi)$: Normalization constant

Theorem 3.2 (Gradient of Log Marginal Likelihood)**.** The gradient with respect to hyperparameter θ_j is:

$$\frac{\partial}{\partial\theta_j}\log p(\mathbf{y}|X, \boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^\top K_y^{-1}\frac{\partial K_y}{\partial\theta_j}K_y^{-1}\mathbf{y} - \frac{1}{2}\operatorname{Tr}\left(K_y^{-1}\frac{\partial K_y}{\partial\theta_j}\right) \tag{34}$$

This enables gradient-based optimization (e.g., L-BFGS, Adam) for hyperparameter tuning.

4 Sparse Gaussian Process Approximations

4.1 Motivation

Exact GP inference scales as $O(N^3)$, prohibitive for large datasets. Sparse methods approximate the full GP using $M \ll N$ *inducing points.*

Key idea: Introduce a set of inducing variables $\mathbf{u} = f(\mathbf{Z})$ at locations $\mathbf{Z} = \{z_1, \dots, z_M\}$, where $M \ll N$. These inducing points act as a summary of the data.

4.2 Inducing Variable Framework

Definition 4.1 (Inducing Variables)**.** Let $\mathbf{Z} \in \mathbb{R}^{M \times D}$ be a set of inducing input locations, and define the inducing variables:

$$\mathbf{u} = [f(z_1), \dots, f(z_M)]^\top \sim \mathcal{N}(\mathbf{0}, K_{uu}) \tag{35}$$

where $K_{uu} = K(\mathbf{Z}, \mathbf{Z}) \in \mathbb{R}^{M \times M}$.

The joint distribution of inducing variables, training function values, and test function values is:

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K_{uu} & K_{uf} & K_{u*} \\ K_{fu} & K_{ff} & K_{f*} \\ K_{*u} & K_{*f} & K_{**} \end{bmatrix}\right) \tag{36}$$

4.3 Deterministic Training Conditional (DTC) Approximation

The DTC approximation assumes that conditioned on $\mathbf{u}$, the training and test function values are independent:

$$p(\mathbf{f}, \mathbf{f}_* | \mathbf{u}) = p(\mathbf{f}|\mathbf{u}) p(\mathbf{f}_*|\mathbf{u}) \tag{37}$$

This leads to the approximate prior:

$$q(\mathbf{f}) = \int p(\mathbf{f}|\mathbf{u}) p(\mathbf{u}) d\mathbf{u} \tag{38}$$

Using Gaussian conditioning:

$$p(\mathbf{f}|\mathbf{u}) = \mathcal{N}(K_{fu} K_{uu}^{-1} \mathbf{u}, K_{ff} - Q_{ff}) \tag{39}$$

$$Q_{ff} = K_{fu} K_{uu}^{-1} K_{uf} \tag{40}$$

Under the DTC approximation, we replace K_{ff} with Q_{ff}:

$$q_{\text{DTC}}(\mathbf{f}) = \mathcal{N}(K_{fu} K_{uu}^{-1} \mathbf{u}, 0) = \delta(K_{fu} K_{uu}^{-1} \mathbf{u}) \tag{41}$$

This is a *deterministic* mapping from $\mathbf{u}$ to $\mathbf{f}$.

4.4 Variational Free Energy (VFE) / FITC Approximation

To address the overly confident predictions of DTC, the Fully Independent Training Conditional (FITC) approximation retains the diagonal of the residual covariance:

$$q_{\text{FITC}}(\mathbf{f}) = \mathcal{N}(K_{fu} K_{uu}^{-1} \mathbf{u}, \operatorname{diag}[K_{ff} - Q_{ff}]) \tag{42}$$

This preserves marginal variances while assuming conditional independence across data points given $\mathbf{u}$.

4.5 Sparse Variational Gaussian Processes (SVGP)

The most general and principled sparse approximation is the *Sparse Variational GP*, which treats the inducing variables as variational parameters.

4.5.1 Variational Inference Framework

Given data $\mathbf{y}$, we want to compute the posterior $p(\mathbf{f}, \mathbf{u}|\mathbf{y})$. Exact inference is intractable, so we introduce a variational distribution $q(\mathbf{f}, \mathbf{u})$ and minimize the KL divergence:

$$q^*(\mathbf{f}, \mathbf{u}) = \arg\min_q \mathrm{KL}(q(\mathbf{f}, \mathbf{u}) \| p(\mathbf{f}, \mathbf{u}|\mathbf{y})) \tag{43}$$

4.5.2 Evidence Lower Bound (ELBO)

Theorem 4.1 (ELBO Derivation)**.** The log marginal likelihood decomposes as:

$$\log p(\mathbf{y}) = \mathcal{L}(q) + \mathrm{KL}(q(\mathbf{f}, \mathbf{u}) \| p(\mathbf{f}, \mathbf{u}|\mathbf{y})) \tag{44}$$

where the Evidence Lower Bound (ELBO) is:

$$\mathcal{L}(q) = \mathbb{E}_{q(\mathbf{f},\mathbf{u})}[\log p(\mathbf{y}|\mathbf{f})] - \mathrm{KL}(q(\mathbf{f}, \mathbf{u}) \| p(\mathbf{f}, \mathbf{u})) \tag{45}$$

Proof. Starting from Bayes' rule:

$$\log p(\mathbf{y}) = \log p(\mathbf{y}|\mathbf{f}, \mathbf{u}) + \log p(\mathbf{f}, \mathbf{u}) - \log p(\mathbf{f}, \mathbf{u}|\mathbf{y}) \tag{46}$$

Taking expectation under $q(\mathbf{f}, \mathbf{u})$:

$$\begin{aligned}
\log p(\mathbf{y}) &= \mathbb{E}_q[\log p(\mathbf{y}|\mathbf{f})] + \mathbb{E}_q[\log p(\mathbf{f}, \mathbf{u})] - \mathbb{E}_q[\log p(\mathbf{f}, \mathbf{u}|\mathbf{y})] && (47)\\
&= \mathbb{E}_q[\log p(\mathbf{y}|\mathbf{f})] - \mathrm{KL}(q(\mathbf{f}, \mathbf{u}) \| p(\mathbf{f}, \mathbf{u})) - \mathbb{E}_q[\log q(\mathbf{f}, \mathbf{u})] && (48)\\
&\quad + \mathbb{E}_q[\log p(\mathbf{f}, \mathbf{u}|\mathbf{y})] && (49)\\
&= \underbrace{\mathbb{E}_q[\log p(\mathbf{y}|\mathbf{f})] - \mathrm{KL}(q(\mathbf{f}, \mathbf{u}) \| p(\mathbf{f}, \mathbf{u}))}_{\mathcal{L}(q)} && (50)\\
&\quad + \underbrace{\mathbb{E}_q\left[\log \frac{p(\mathbf{f}, \mathbf{u}|\mathbf{y})}{q(\mathbf{f}, \mathbf{u})}\right]}_{\mathrm{KL}(q\|p)} && (51)
\end{aligned}$$

Since $\mathrm{KL}(q\|p) \geq 0$, we have $\log p(\mathbf{y}) \geq \mathcal{L}(q)$, hence $\mathcal{L}(q)$ is a lower bound. □

4.5.3 Optimal Variational Distribution

We factorize the variational distribution as:

$$q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u})q(\mathbf{u}) \tag{52}$$

where $p(\mathbf{f}|\mathbf{u})$ is the exact GP conditional and $q(\mathbf{u})$ is a free-form variational distribution.

Theorem 4.2 (Optimal $q(\mathbf{u})$)**.** The variational distribution that maximizes the ELBO is Gaussian:

$$q(\mathbf{u}) = \mathcal{N}(\mathbf{m}, S) \tag{53}$$

with mean $\mathbf{m} \in \mathbb{R}^M$ and covariance $S \in \mathbb{R}^{M\times M}$ as variational parameters.

4.5.4 ELBO Computation

Substituting the factorized q into the ELBO:

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{f})}[\log p(\mathbf{y}|\mathbf{f})] - \mathrm{KL}(q(\mathbf{u})\|p(\mathbf{u})) \tag{54}$$

For Gaussian likelihoods $p(y_i|f_i) = \mathcal{N}(y_i|f_i, \sigma_n^2)$:

$$\begin{aligned}\mathcal{L} &= \sum_{i=1}^{N} \mathbb{E}_{q(f_i)}[\log \mathcal{N}(y_i|f_i, \sigma_n^2)] - \mathrm{KL}(q(\mathbf{u})\|p(\mathbf{u})) &(55)\\ &= -\frac{N}{2}\log(2\pi\sigma_n^2) - \frac{1}{2\sigma_n^2}\sum_{i=1}^{N} \mathbb{E}_{q(f_i)}[(y_i - f_i)^2] - \mathrm{KL}(q(\mathbf{u})\|p(\mathbf{u})) &(56)\end{aligned}$$

Expected squared error:

$$\begin{aligned}\mathbb{E}_{q(f_i)}[(y_i - f_i)^2] &= \mathbb{E}_{q(f_i)}[y_i^2 - 2y_i f_i + f_i^2] &(57)\\ &= y_i^2 - 2y_i\,\mathbb{E}_{q(f_i)}[f_i] + \mathbb{E}_{q(f_i)}[f_i^2] &(58)\\ &= y_i^2 - 2y_i\mu_i + (\mu_i^2 + v_i) &(59)\end{aligned}$$

where:

$$\begin{aligned}q(f_i) &= \mathcal{N}(\mu_i, v_i) &(60)\\ \mu_i &= \mathbf{k}_i^\top K_{uu}^{-1}\mathbf{m} &(61)\\ v_i &= k_{ii} - \mathbf{k}_i^\top K_{uu}^{-1}\mathbf{k}_i + \mathbf{k}_i^\top K_{uu}^{-1} S K_{uu}^{-1}\mathbf{k}_i &(62)\end{aligned}$$

and $\mathbf{k}_i = [k(x_i, z_1), \dots, k(x_i, z_M)]^\top$.
KL divergence term:

$$\begin{aligned}\mathrm{KL}(q(\mathbf{u})\|p(\mathbf{u})) &= \mathrm{KL}(\mathcal{N}(\mathbf{m}, S)\|\mathcal{N}(\mathbf{0}, K_{uu})) &(63)\\ &= \frac{1}{2}\left[\mathrm{Tr}(K_{uu}^{-1}S) + \mathbf{m}^\top K_{uu}^{-1}\mathbf{m} - M + \log\frac{|K_{uu}|}{|S|}\right] &(64)\end{aligned}$$

4.5.5 Final ELBO Expression

SVGP ELBO

$$\mathcal{L}(\mathbf{m}, S, \mathbf{Z}, \boldsymbol{\theta}) = -\frac{1}{2\sigma_n^2}\sum_{i=1}^{N}(y_i - \mu_i)^2 - \frac{1}{2\sigma_n^2}\sum_{i=1}^{N} v_i - \mathrm{KL}(q(\mathbf{u})\|p(\mathbf{u})) + \text{const} \tag{65}$$

Optimization: Maximize $\mathcal{L}$ with respect to:

- Variational parameters: $\mathbf{m}, S$
- Inducing locations: $\mathbf{Z}$
- Kernel hyperparameters: $\boldsymbol{\theta}$

4.6 Computational Complexity of SVGP

Per-iteration cost:

- Computing μ_i, v_i for all i: $O(NM^2)$
- KL divergence: $O(M^3)$ (Cholesky of K_{uu})
- Total: $O(NM^2)$ per iteration

Stochastic optimization: Using mini-batches of size $B \ll N$ reduces per-iteration cost to $O(BM^2)$, enabling scaling to millions of data points.

4.7 Predictive Distribution

For a test point x_*, the predictive distribution is:

$$p(f_*|\mathbf{y}, x_*) \approx \int p(f_*|\mathbf{u}, x_*)q(\mathbf{u})d\mathbf{u} \tag{66}$$

$$= \mathcal{N}(f_*|\mu_*, \sigma_*^2) \tag{67}$$

where:

$$\mu_* = \mathbf{k}_*^\top K_{uu}^{-1}\mathbf{m} \tag{68}$$

$$\sigma_*^2 = k_{**} - \mathbf{k}_*^\top K_{uu}^{-1}\mathbf{k}_* + \mathbf{k}_*^\top K_{uu}^{-1} S K_{uu}^{-1}\mathbf{k}_* \tag{69}$$

and $\mathbf{k}_* = [k(x_*, z_1), \ldots, k(x_*, z_M)]^\top$.

Remark 4.1. Prediction cost is $O(M^2)$ per test point, independent of N.

5 Implementation Details

5.1 Inducing Point Initialization

Common strategies:

1. **Random subset**: Sample M points uniformly from training data
2. **K-means clustering**: Use cluster centers as inducing points
3. **Greedy variance maximization**: Iteratively add points that maximize predictive variance
4. **Grid-based**: Place points on a regular grid (works for low-dimensional inputs)

In our implementation, we use random initialization and allow gradient-based optimization to refine locations.

5.2 Parameterization Tricks

1. Cholesky Parameterization of S: To ensure $S \succ 0$, parameterize:

$$S = LL^\top \tag{70}$$

where $L \in \mathbb{R}^{M\times M}$ is lower triangular. Optimize over the entries of L.

2. Noise Constraint: Constrain $\sigma_n \in [\epsilon_{\min}, \epsilon_{\max}]$ to prevent numerical instability:

$$\sigma_n = \epsilon_{\min} + (\epsilon_{\max} - \epsilon_{\min}) \cdot \text{sigmoid}(\tilde{\sigma}_n) \tag{71}$$

3. Log-space for positive parameters: For lengthscales ℓ and variances σ_f^2, optimize in log-space:

$$\ell = \exp(\log \ell), \quad \sigma_f^2 = \exp(\log \sigma_f^2) \tag{72}$$

5.3 Stochastic Optimization

Algorithm 1 Stochastic Variational Inference for Sparse GP

```
1: Initialize $\mathbf{m}, L, \mathbf{Z}, \boldsymbol{\theta}$
2: **for** epoch = 1 to $T$ **do**
3:     **for** mini-batch $\mathcal{B} \subset \{1, \dots, N\}$ **do**
4:         Compute $\mu_i, v_i$ for $i \in \mathcal{B}$
5:         Compute scaled ELBO: $\tilde{\mathcal{L}} = \frac{N}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \log p(y_i|f_i) - \mathrm{KL}(q(\mathbf{u})\|p(\mathbf{u}))$
6:         Compute gradients: $\nabla_{\mathbf{m}}\tilde{\mathcal{L}}, \nabla_L\tilde{\mathcal{L}}, \nabla_{\mathbf{Z}}\tilde{\mathcal{L}}, \nabla_{\boldsymbol{\theta}}\tilde{\mathcal{L}}$
7:         Update parameters using Adam/SGD
8:     **end for**
9: **end for**
```

5.4 Numerical Stability

1. Jitter for matrix inversion: Add small diagonal term to ensure positive definiteness:

$$K_{uu} \leftarrow K_{uu} + \lambda I, \quad \lambda \approx 10^{-6} \tag{73}$$

2. Use Cholesky decomposition instead of direct inversion: To compute $K_{uu}^{-1}\mathbf{v}$, solve:

$$K_{uu}\mathbf{x} = \mathbf{v} \tag{74}$$

using forward-backward substitution on the Cholesky factor $LL^\top = K_{uu}$.

3. Log-determinant computation:

$$\log|K_{uu}| = 2\sum_{i=1}^{M} \log L_{ii} \tag{75}$$

where L is the Cholesky factor.

6 Application to Residual Modeling

6.1 Two-Stage Approach

Our system employs a two-stage prediction architecture:

Stage 1: Baseline Predictor (LightGBM)

$$\hat{y}_{\text{base}}(x) = \text{LGBM}(x; \theta_{\text{lgbm}}) \tag{76}$$

Stage 2: Residual Uncertainty (Sparse GP)

$$r_i = y_i - \hat{y}_{\text{base}}(x_i) \tag{77}$$

$$r(x) \sim \mathcal{GP}(0, k(x, x')) \tag{78}$$

6.2 Rationale

1. **Computational efficiency**: LightGBM handles mean prediction efficiently via gradient boosting
2. **Heteroscedastic uncertainty**: GP models input-dependent residual variance
3. **Calibration**: GP naturally provides calibrated uncertainty estimates
4. **Modularity**: Stages can be developed and debugged independently

6.3 Final Prediction

For a test input x_*:

$$\mu_*(x_*) = \hat{y}_{\text{base}}(x_*) + \mu_{\text{GP}}(x_*) \tag{79}$$

$$\sigma_*(x_*)^2 = \sigma_{\text{GP}}(x_*)^2 + \sigma_n^2 \tag{80}$$

The predictive distribution is:

$$p(y_*|x_*, \mathcal{D}) = \mathcal{N}(y_*|\mu_*(x_*), \sigma_*(x_*)^2) \tag{81}$$

7 Advanced Topics

7.1 Multi-Output Gaussian Processes

For vector-valued functions $\mathbf{f}(x) = [f_1(x), \ldots, f_P(x)]^\top$, we can use:

1. Independent GPs:

$$f_p(x) \sim \mathcal{GP}(0, k_p(x, x')) \tag{82}$$

2. Linear Model of Coregionalization (LMC):

$$k_{\text{LMC}}((x, p), (x', p')) = \sum_{q=1}^{Q} B^{(q)}_{pp'} k_q(x, x') \tag{83}$$

where $B^{(q)} \in \mathbb{R}^{P \times P}$ are positive semi-definite matrices.

7.2 Sparse Spectrum Gaussian Processes

An alternative to inducing points is to approximate the kernel in the frequency domain:

$$k(x, x') \approx \frac{1}{M} \sum_{m=1}^{M} \cos(\omega_m^\top x + b_m) \cos(\omega_m^\top x' + b_m) \tag{84}$$

where $\omega_m \sim p(\omega)$ from the kernel's spectral density.

7.3 Deep Gaussian Processes

Hierarchical composition of GPs:

$$\mathbf{f}_1(x) \sim \mathcal{GP}(0, k_1(x, x')) \tag{85}$$
$$\mathbf{f}_2(\mathbf{f}_1) \sim \mathcal{GP}(0, k_2(\mathbf{f}_1, \mathbf{f}_1')) \tag{86}$$
$$\vdots \tag{87}$$
$$y = \mathbf{f}_L(\mathbf{f}_{L-1}) + \epsilon \tag{88}$$

Provides greater expressiveness but requires doubly-stochastic variational inference.

7.4 Non-Gaussian Likelihoods

For classification ($y \in \{0, 1\}$) or count data, use:

- **Bernoulli**: $p(y|f) = \text{Bernoulli}(\sigma(f))$ where σ is the logistic function
- **Poisson**: $p(y|f) = \text{Poisson}(\exp(f))$

The ELBO expectation $\mathbb{E}_{q(f)}[\log p(y|f)]$ becomes intractable and requires:

- Monte Carlo sampling
- Analytic approximations (e.g., probit approximation)
- Quadrature methods

8 Conclusion

This document has provided a comprehensive mathematical foundation for understanding Gaussian processes and their sparse variational approximations. Key takeaways:

1. **Function-space view**: GPs place priors directly over functions, enabling principled uncertainty quantification
2. **Exact inference**: GP regression provides closed-form posteriors but scales as $O(N^3)$
3. **Sparse approximations**: Inducing variables reduce complexity to $O(NM^2)$ with minimal loss of accuracy

4. **Variational inference**: The ELBO framework provides a principled objective for learning inducing locations and hyperparameters
5. **Two-stage modeling**: Combining gradient boosting with GP residuals balances efficiency and uncertainty calibration

The mathematical rigor developed here underpins the uncertainty-aware forecasting system, enabling:

- Calibrated prediction intervals
- Automatic out-of-distribution detection via uncertainty inflation
- Risk-aware decision-making through expected value calculations

Further Reading

1. Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.
2. Hensman, J., Fusi, N., & Lawrence, N. D. (2013). Gaussian processes for big data. *UAI*.
3. Titsias, M. (2009). Variational learning of inducing variables in sparse Gaussian processes. *AISTATS*.
4. Matthews, A. G. D. G., et al. (2017). GPflow: A Gaussian process library using TensorFlow. *JMLR*.
5. Burt, D. R., et al. (2020). Convergence of sparse variational inference in Gaussian processes regression. *JMLR*.