Developing a Sparse and Scalable Gaussian Process framework for Bioprocesses

Author: Jakub Polák
Affiliation: DataHow AG, R&D Team
Scope: Internship
Skills: Strong ML background, Python/PyTorch, software engineering & benchmarking
Date: September 26, 2025
Contact: h.narayanan@datahow.ch

Executive Summary

Hybrid bioprocess models that discretize system dynamics and augment mechanistic knowledge with machine learning are highly effective in low-data regimes. Within our framework, Gaussian Processes (GPs) model stepwise deviations/growth rates and capture correlations between observations, yielding a predictive model with uncertainty. However, exact GP training and inference scale cubically in the number of observations, which becomes prohibitive as we pool more experiments and longer trajectories.

This project will extend our existing gpytorch-based GP module with sparse and scalable alternatives that preserve predictive performance while dramatically improving memory and runtime scalability.

Background

Brief introduction

Bioprocess modeling benefits from hybrid approaches that combine kinetics-based knowledge, in the form of Ordinary Differential Equations (ODEs), with data-driven corrections. In our stepwise/discretized hybrid model pipeline, each observation provides a training sample for a GP that models residual dynamics or unmodeled effects. This delivers strong predictions with uncertainty estimations when data are scarce and heterogeneous. As we integrate more experimental campaigns, the $O(n^3)$ cost of exact GPs in training and $O(n^2)$ memory become the bottleneck, limiting both research iteration speed and industrial deployment.

State of the art

Scalable GPs can reduce complexity via approximations or linear algebra. The project will include a literature review and theoretical analysis into which approaches would be suitable. Some investigation suggestions include existing scalable GPs python libraries and proposed modification such as inducing-point variational methods (SVGP/VFE), structured kernel interpolation (SKI/KISS-GP), stochastic optimization with mini-batches, preconditioned conjugate gradient (BBMM) solvers, and fast predictive covariance approximations (LOVE). We will benchmark these approaches across internal in-silico and real bioprocess datasets under realistic scaling scenarios, reporting accuracy–efficiency trade-offs and deployment readiness. The outcome will be

a practical, production-ready GP component suitable for industrial hybrid modeling at larger data scales.

Challenges

- Scale along the experiment axis: n grows primarily with the number of experiments/trajectories; from current hundreds, we must be able to handle thousands to hundreds of thousands of observations without losing uncertainty calibration.
- Hyperparameter & inducing set selection: Choosing m, placing inducing points/features, and stabilizing optimization (whitening, natural gradients, preconditioning) are non-trivial and data-dependent.
- Irregular sampling & multi-output settings: Bioprocess time series are irregular, multi-sensor, and partially observed; kernels and batching must respect this.
- Deployment constraints: Inference latency, memory footprint matter for on-prem/edge use.

Approach

We will augment our current gpytorch GP module with interchangeable scalable back-ends and a rigorous benchmarking harness. The focus is to retain predictive performance and calibrated uncertainty while lowering compute and memory to enable larger study sizes.

Objectives

1. **Scalable GP module in gpytorch:** Drop-in replacement for the current GP with back-ends (Exact, SVGP/VFE, SKI/KISS-GP, Exact+BBMM) and unified API.
2. **Benchmark suite & scaling curves:** End-to-end benchmarks on internal datasets. Produce accuracy–efficiency Pareto plots vs. number of experiments n and inducing size m.
3. **Uncertainty evaluation:** Calibration of prediction interval, sharpness, negative log-likelihood and decision-centric metrics relevant for hybrid control.
4. **Deployment report:** Latency and memory profiling for batch and single-point predictions; guidance for method selection by regime (n, d, noise, irregularity).
5. **Documentation & handover:** Clean code, examples, and a concise playbook for practitioners.

Methods and Work Plan

Data and Resources

- Datasets: Existing in-silico bioprocess datasets with varying size, noise and sampling; selected de-identified real processes for external validity (subject to availability).

- Software: Python, PyTorch, gpytorch; optional PyKeOps for large kernel ops; Hydra/Lightning for experiment orchestration; MLFlow / W&B (or equivalent) for tracking.
- Compute: Remote connection server available for experiment execution; profiling on both CPU and GPU.

Timeline

Timeline subject to the scope. Preference: Longer the better.

Expected Outcome

A production-ready scalable GP module integrated into our hybrid pipeline. Benchmark results & Pareto frontiers demonstrating comparable accuracy to Exact GP with reductions in memory/time (dataset-dependent), along with calibrated uncertainty. A practitioner's guide recommending when to use exact or scalable GPs approaches for typical bioprocess settings (n, d, sampling patterns).

References

- Snelson, E., & Ghahramani, Z. (2006). Sparse GP using Pseudo-inputs (SPGP). NeurIPS.
- Hensman, J., Fusi, N., & Lawrence, N. (2013). Gaussian Processes for Big Data (SVGP/Stochastic VI).
- Wilson, A. G., & Nickisch, H. (2015). Kernel Interpolation for Scalable Structured GPs (KISS-GP/SKI). ICML.
- Gardner, J. R., Pleiss, G., Bindel, D., Weinberger, K. Q., & Wilson, A. G. (2018). GPyTorch: Blackbox Matrix-Matrix Inference (BBMM). NeurIPS
- Hensman, J., Durrande, N., & Solin, A. (2018). Variational Fourier Features for GPs. JMLR.
- Pleiss, G., Gardner, J. R., Weinberger, K. Q., & Wilson, A. G. (2018). Constant-Time Predictive Distributions for GPs (LOVE). ICML.
- Burt, D. R., Rasmussen, C. E., & van der Wilk, M. (2020). Convergence of Sparse Variational Inference in GPs. JMLR.