POPS for Gaussian Processes: A Rigorous Report

Automated report

November 5, 2025

Abstract

This report documents the Gaussian-process-aware implementation of Algorithm 1 (POPS) used in the notebook $\mathrm{gp}_{p}ops.ipynb.Itgivesprecisenotation, showsthealgorithmicmappingtoreproducing-kernelHilbertspace(RKHS)minimal-normcorrections, provesthattheresultingpointwiseenvelopeenclosesalltrainir$ $containedandintendedtobeusedasareferenceforexperimentsandfurtherdevelopment.$

1 Problem setup and notation

We observe training inputs $X = \{x_i\}_{i=1}^n \subset \mathcal{X}$ and noisy scalar responses $y \in \mathbb{R}^n$. A prior Gaussian process (GP) with zero mean and covariance kernel $k(\cdot, \cdot)$ is used for prediction. Kernel hyperparameters (ℓ, σ_f, etc.) are fixed for this discussion.

Define

$$K_{XX} = [k(x_i, x_j)]_{i,j=1}^n, \qquad K = K_{XX} + \sigma_n^2 I_n,$$

and let the usual posterior weight vector be

$$\alpha = K^{-1} y \quad \text{(computed stably via Cholesky)}.$$

For a test grid X_{test} with kernel cross-covariances $K_{X_{\text{test}},X}$, the standard GP predictive mean is

$$\mu(x) = k(x, X)\, \alpha, \qquad \mu_{\text{train}} = K_{XX}\, \alpha.$$

2 POPS idea (function-space corrections)

POPS (post hoc output-preserving set) seeks to construct an envelope of functions that (i) are close to the GP posterior mean in RKHS norm sense, and (ii) are guaranteed to include the observed targets at the training inputs. The algorithm implemented constructs a set of candidate functions by applying single-point minimal-norm kernel corrections centered at each training input, then forms pointwise lower/upper envelopes of these candidates.

For each training index i:

- Compute the residual at training input x_i:

$$\delta_i = y_i - \mu_{\text{train}}(x_i) = y_i - \mu_{\text{train,i}}.$$

- Let $k_{ii} = k(x_i, x_i)$ be the diagonal of K_{XX}. Define the scalar correction coefficient

$$\beta_i = \frac{\delta_i}{k_{ii}}.$$

- Define the correction function
$$c_i(\cdot) = \beta_i\, k(\cdot, x_i).$$
- Form the candidate corrected mean function
$$\mu^{(i)}(\cdot) = \mu(\cdot) + c_i(\cdot).$$

Finally define pointwise envelopes on the test grid:

$$\mu_{\min}(x) = \min_{i=1,\dots,n} \mu^{(i)}(x), \qquad \mu_{\max}(x) = \max_{i=1,\dots,n} \mu^{(i)}(x).$$

3 Why the single-point corrections are minimal-norm in RKHS

Let $\mathcal{H}_k$ denote the RKHS associated with k. The representer theorem and reproducing property yield that any function $h \in \mathcal{H}_k$ with value $h(x_i) = \delta_i$ and minimal RKHS norm must be a scalar multiple of the kernel section $k(\cdot, x_i)$. Formally, among all h satisfying $h(x_i) = \delta_i$,

$$\|h\|^2_{\mathcal{H}_k} \geq \frac{\delta_i^2}{k_{ii}},$$

with equality attained by $h(\cdot) = \frac{\delta_i}{k_{ii}} k(\cdot, x_i)$. Proof sketch: by the reproducing property,

$$\delta_i = h(x_i) = \langle h, k(\cdot, x_i)\rangle_{\mathcal{H}_k},$$

so by Cauchy–Schwarz, $|\delta_i| \leq \|h\|_{\mathcal{H}_k} \|k(\cdot, x_i)\|_{\mathcal{H}_k}$, and $\|k(\cdot, x_i)\|^2_{\mathcal{H}_k} = k_{ii}$. Rearranging gives the lower bound on $\|h\|_{\mathcal{H}_k}$ and equality for the stated multiple.

Therefore each correction c_i is the minimal-RKHS-norm function that adjusts the posterior mean so that the corrected mean attains the observed y_i at x_i.

4 Correctness: envelope encloses training targets

At a training input x_j, evaluate candidate $\mu^{(i)}$:

$$\mu^{(i)}(x_j) = \mu_{\text{train}}(x_j) + \beta_i k(x_j, x_i).$$

In particular, for $i = j$,

$$\mu^{(j)}(x_j) = \mu_{\text{train}}(x_j) + \beta_j k_{jj} = \mu_{\text{train}}(x_j) + \delta_j = y_j.$$

Hence among the n candidates $\{\mu^{(i)}\}$ the one with $i = j$ attains exactly y_j at x_j. Consequently,

$$\min_i \mu^{(i)}(x_j) \leq y_j \leq \max_i \mu^{(i)}(x_j),$$

so both envelopes $\mu_{\min}, \mu_{\max}$ enclose the observed training target y_j at every training input. This is the basic correctness property required by POPS.

Note: the inclusion is exact at training inputs (up to numerical tolerance), because one candidate reproduces the target exactly; the envelope may be looser away from training points.

5 Discussion of assumptions and numerical issues

- Kernel diagonal nonzero: RBF and most common kernels have $k_{ii} > 0$. If k_{ii} is numerically tiny, compute $\beta_i = \delta_i/(k_{ii} + \varepsilon)$ with a small regulariser ε to avoid blow-up.
- Use of noise in inference vs correction: the posterior mean μ is computed using the inference covariance $K = K_{XX} + \sigma_n^2 I$ (so $\alpha = K^{-1}y$). Corrections use the noise-free kernel diagonal k_{ii} (since correction is an RKHS operation). This choice is consistent with the interpretation that observation noise affects posterior estimation but pointwise function constraints are enforced in the underlying function space.
- Numerical stability: compute α via Cholesky solves (as in the notebook). When forming many corrections and evaluating at a dense test grid cost and memory grow: storing the full $n \times n_{\text{test}}$ matrix of candidates is O(n n_{test}); $consider streaming, computing envelopes online, or evaluating corrections so$
- Floating tolerance: test that targets lie within envelopes with a small tolerance (e.g. 10^{-12} or a problem-dependent tolerance).

6 Computational complexity and practical scaling

Let n be training size and m the number of test points.

- Kernel matrix formation: $O(n^2)$ memory, $O(n^2)$ compute for entries.
- Cholesky of K: $O(n^3)$ time.
- Forming every correction evaluated on m test points: $O(nm)$ to compute each kernel column; total O(n m) per loop and O($n^2 m$) $if naively recomputing redundant elements. The notebook computes kernel(X$ n candidates. Memory to hold candidates is O(m n).

For large n: use sparse/approximate GP methods (inducing points, SKI, local GP, random Fourier features) or reduce the number of candidates by sub-sampling influential training points.

7 Variants and extensions

- Sequential POPS: apply corrections sequentially, updating the mean after each enforced constraint and recomputing the minimal-norm next correction that respects previous enforced constraints. This yields a path of minimal-norm updates that jointly enforce multiple constraints in a greedy fashion; it is more expensive but produces a different envelope (often tighter).
- Joint minimal-norm interpolation: to enforce all constraints exactly while minimising RKHS norm you would solve a linear system in the RKHS (equivalently choose coefficients γ for basis $k(\cdot, x_i)$ that satisfy $\mu_{\text{train}} + K_{XX}\gamma = y$ and minimize $\|\mu - (\mu + \sum \gamma_i k(\cdot, x_i))\|_{\mathcal{H}_k}$). That yields a single function but may have larger norm than the per-point minimal corrections.
- Probabilistic envelopes: adopt uncertainty-aware bounds using posterior covariance; POPS as implemented enforces deterministic inclusion at training points only.

8 Validation used in code

The accompanying test in $\mathrm{gp}_{p}ops.ipynbbuildsthecandidatematrixattraininginputsandchecks, perindexj$, that

$$\min_i \mu^{(i)}(x_j) \le y_j \le \max_i \mu^{(i)}(x_j)$$

to numerical tolerance. The test passes because $\mu^{(j)}(x_j) = y_j$, as shown above.

9 Conclusions

The implemented GP-aware POPS:

- Interprets POPS corrections as RKHS minimal-norm single-point adjustments.
- Ensures by construction that every training target is included in the pointwise envelope.
- Is straightforward, interpretable, and computationally simple for moderate n, but requires approximations to scale to very large datasets.

Practical commands

To reproduce results and check enclosure:

- Use Cholesky solves for numerical stability.
- Regularise kernel diagonals if needed, and use small tolerances when asserting inclusion.

References and further reading:

- RKHS basics and representer theorem: Schölkopf and Smola (2002), "Learning with Kernels".
- Gaussian processes: Rasmussen and Williams (2006), "Gaussian Processes for Machine Learning".
- POPS original material: see the user's POPS Algorithm 1 reference (pops.pdf).