Preconditioning for Scalable Gaussian Process Hyperparameter Optimization

Jonathan Wenger Geoff Pleiss Philipp Hennig John P. Cunningham Jacob R. Gardner

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

MAX PLANCK INSTITUTE FOR INTELLIGENT SYSTEMS

erc

Penn UNIVERSITY of PENNSYLVANIA

COLUMBIA | Zuckerman Institute

MORTIMER B. ZUCKERMAN MIND BRAIN BEHAVIOR INSTITUTE

In A Nutshell

Preconditioning can be exploited for highly efficient log-determinant estimation and in turn GP hyperparameter optimization.

Goal: Large-scale Gaussian process hyperparameter optimization.

Known: Can be reduced to matrix-vector multiplication. [1–7]

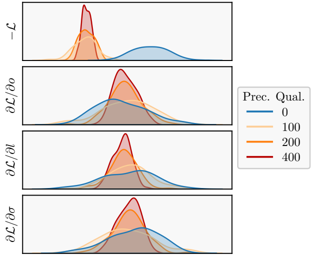
Problem: Stochastic trace estimates of $\log \det(\hat{\boldsymbol{K}})$ and its gradient.

- Require many random vectors to converge.
- Introduce stochasticity into optimization.

$\Longrightarrow$ slows down training

Our work: Precondition stochastic trace estimators.

- Preconditioning can be used to reduce variance – i.e. accelerate convergence.
- Theoretical guarantees for all approximations.
- Practical preconditioner choices for given kernels.
- Up to twelvefold training speedup.



Large-scale GP Hyperparameter Optimization

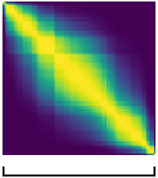
A numerical linear algebra bottleneck.

Need to: Evaluate log-marginal likelihood and its derivative repeatedly.

Challenge: Computationally costly operations with the kernel matrix.

- linear solves $\boldsymbol{v} \mapsto \hat{\boldsymbol{K}}^{-1}\boldsymbol{v}$
- matrix traces $\log\det(\hat{\boldsymbol{K}}) = \operatorname{tr}(\log(\hat{\boldsymbol{K}}))$ and $\operatorname{tr}\left(\hat{\boldsymbol{K}}^{-1}\frac{\partial \hat{\boldsymbol{K}}}{\partial \boldsymbol{\theta}_i}\right)$

$\hat{\boldsymbol{K}} =$



$n \times n$

Linear solves and matrix traces can be computed solely via *matrix-vector multiplication*! [4, 5, 8]

This is great because ...

- matrix-vector multiplies have complexity $\mathcal{O}(n^2)$.
- structured or sparse matrices are efficient to multiply with.
- the kernel matrix does not need to be stored in memory explicitly [9].
- we can exploit parallelization and modern hardware (GPUs) [5].

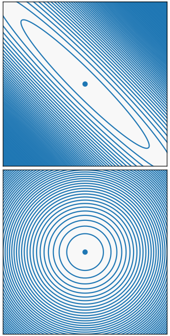
lower time and space complexity

Preconditioning

How to encode and leverage structural prior knowledge about matrices.

Preconditioner

$$\hat{\boldsymbol{P}} \approx \hat{\boldsymbol{K}}$$

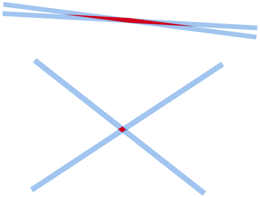


such that $\kappa(\hat{\boldsymbol{P}}^{-1}\hat{\boldsymbol{K}}) \ll \kappa(\hat{\boldsymbol{K}})$ and $\hat{\boldsymbol{P}}$ is computationally tractable.

- Computing and storing $\hat{\boldsymbol{P}}$ is cheap.
- Linear solves $\boldsymbol{v} \mapsto \hat{\boldsymbol{P}}^{-1}\boldsymbol{v}$ are efficient.
- Derived properties, such as the determinant or spectrum are known.

Asymptotic approx. error $g(\ell) \to 0$ of sequence of preconditioners $\hat{\boldsymbol{P}}_\ell \to \hat{\boldsymbol{K}}$:

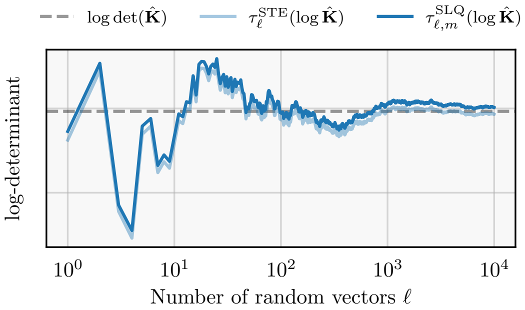
$$\kappa(\hat{\boldsymbol{P}}_\ell^{-1}\hat{\boldsymbol{K}}) \leq (1 + \mathcal{O}(g(\ell))\|\hat{\boldsymbol{K}}\|_F)^2$$



Known Use: Accelerate and stabilize linear solves via CG $\Rightarrow$ bias reduction

Stochastic Trace Estimation

Computing matrix traces $\mathrm{tr}(f(\hat{\boldsymbol{K}}))$ via matrix-vector multiplication [4, 10, 11].



$$
\begin{aligned}
\mathrm{tr}(f(\hat{\boldsymbol{K}})) &= n\mathbb{E}[\boldsymbol{z}_i^\intercal f(\hat{\boldsymbol{K}})\boldsymbol{z}_i] \\
&\approx \tau_\ell^{\mathrm{STE}}(f(\hat{\boldsymbol{K}})) = \frac{n}{\ell}\sum_{i=1}^{\ell} \boldsymbol{z}_i^\intercal f(\hat{\boldsymbol{K}})\boldsymbol{z}_i \\
&\approx \tau_{\ell,m}^{\mathrm{SLQ}}(f(\hat{\boldsymbol{K}}))
\end{aligned}
$$

Problems:

- Worst-case convergence in the number of random vectors is $\mathcal{O}(\ell^{-\frac{1}{2}})$
- Introduces stochasticity into hyperparameter optimization

$\Longrightarrow$ slows down training

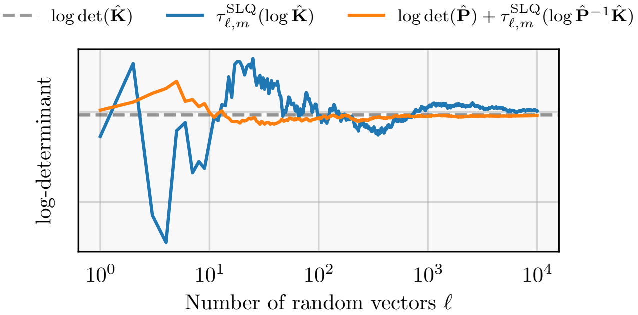
Preconditioned Log-Determinant Estimation

Variance-reduced stochastic trace estimation via preconditioning.

Idea: Decompose $\log$-determinant into deterministic and stochastic approximation.

$$\log\det(\hat{\boldsymbol{K}}) = \log\det\left(\hat{\boldsymbol{P}}_\ell \hat{\boldsymbol{P}}_\ell^{-1}\hat{\boldsymbol{K}}\right) = \underbrace{\log\det(\hat{\boldsymbol{P}}_\ell)}_{\text{known}} + \underbrace{\operatorname{tr}(\log(\hat{\boldsymbol{K}}) - \log(\hat{\boldsymbol{P}}_\ell))}_{\approx \text{ stochastic trace estimate}}$$

The better the preconditioner, the smaller the stochastic approximation $\Rightarrow$ variance reduction



- Backward pass analogously via automatic differentiation.
- If we compute a preconditioner for CG, we can simply reuse it at negligible overhead.
- If $\hat{\boldsymbol{P}}_\ell \to \hat{\boldsymbol{K}}$ at rate $g(\ell)$, then the STE only requires $\mathcal{O}(\ell^{-\frac{1}{2}} g(\ell))$ random vectors.

Convergence Rates for Kernel – Preconditioner Combinations

The faster the preconditioner converges to the kernel matrix (i.e. $g(\ell) \to 0$) the fewer random vectors are needed.

If $\hat{\boldsymbol{P}}_\ell \to \hat{\boldsymbol{K}}$ at rate $g(\ell)$, then the STE only requires $\mathcal{O}(\ell^{-\frac{1}{2}} g(\ell))$ random vectors.

Kernel	d	Preconditioner	$g(\ell)$	Condition
any	$\mathbb{N}$	none	1	
...	...	...	...	...
any	$\mathbb{N}$	RFF	$\ell^{-\frac{1}{2}}$	w/ high probability
RBF	1	partial Cholesky	$\exp(-c\ell)$	for some $c > 0$
RBF	$\mathbb{N}$	QFF	$\exp(-b\ell^{\frac{1}{d}})$	for some $b > 0$ if $\ell^{\frac{1}{d}} > 2\gamma^{-2}$
Matérn(ν)	$\mathbb{N}$	partial Cholesky	$\ell^{-(\frac{2\nu}{d}+1)}$	$2\nu \in \mathbb{N}$ and maximin ordering
Matérn(ν)	1	QFF	$\ell^{-(s(\nu)+1)}$	where $s(\nu) \in \mathbb{N}$
mod. Matérn(ν)	$\mathbb{N}$	QFF	$\ell^{-\frac{s(\nu)+1}{d}}$	where $s(\nu) \in \mathbb{N}$
additive	$\mathbb{N}$	any	$dg(\ell)$	all summands have rate $g(\ell)$
any	$\mathbb{N}$	any kernel approx.	$g(\ell)$	$\exists$ uniform convergence bound

Theoretical Guarantees

Probabilistic error bounds for the estimates of the log-marginal likelihood and its derivative.

Theorem (Log-marginal likelihood)

[…] Then with probability $1-\delta$, the error in the estimate η of the $\log$-marginal likelihood $\mathcal{L}$ satisfies

$$|\eta - \mathcal{L}| \leq \varepsilon_{\mathrm{CG}} + \tfrac{1}{2}(\varepsilon_{\mathrm{Lanczos}} + \varepsilon_{\mathrm{STE}}) \|\log(\hat{\boldsymbol{K}})\|_F,$$

where the individual errors are bounded by

$$\varepsilon_{\mathrm{CG}}(\kappa, m) \leq K_3 \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^m \quad (1)$$

$$\varepsilon_{\mathrm{Lanczos}}(\kappa, m) \leq K_1 \left(\frac{\sqrt{2\kappa+1}-1}{\sqrt{2\kappa+1}+1}\right)^{2m} \quad (2)$$

$$\boxed{\varepsilon_{\mathrm{STE}}(\delta, \ell) \leq C_1 \sqrt{\log(\delta^{-1})}\ell^{-\frac{1}{2}} g(\ell)} \quad (3)$$

Theorem (Derivative)

[…] Then with probability $1-\delta$, the error in the estimate ϕ of the derivative of the $\log$-marginal likelihood $\frac{\partial}{\partial\theta}\mathcal{L}$ satisfies

$$\left|\phi - \tfrac{\partial}{\partial\theta}\mathcal{L}\right| \leq \varepsilon_{\mathrm{CG}} + \tfrac{1}{2}(\varepsilon_{\mathrm{CG}'} + \varepsilon_{\mathrm{STE}}) \left\|\hat{\boldsymbol{K}}^{-1}\frac{\partial\hat{\boldsymbol{K}}}{\partial\theta}\right\|_F$$

where the individual errors are bounded by

$$\varepsilon_{\mathrm{CG}}(\kappa, m) \leq K_4 \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^m \quad (4)$$

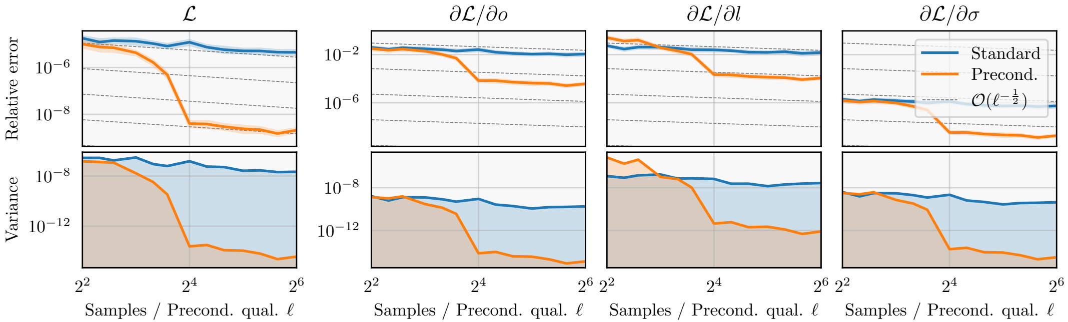
$$\varepsilon_{\mathrm{CG}'}(\kappa, m) \leq K_2 \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^m \quad (5)$$

$$\boxed{\varepsilon_{\mathrm{STE}}(\delta, \ell) \leq C_1 \sqrt{\log(\delta^{-1})}\ell^{-\frac{1}{2}} g(\ell)} \quad (6)$$

We leverage preconditioning not only to reduce bias, but crucially also to reduce variance.

Preconditioning Reduces Bias and Variance

Estimating the $\log$-marginal likelihood and its derivatives on synthetic data.

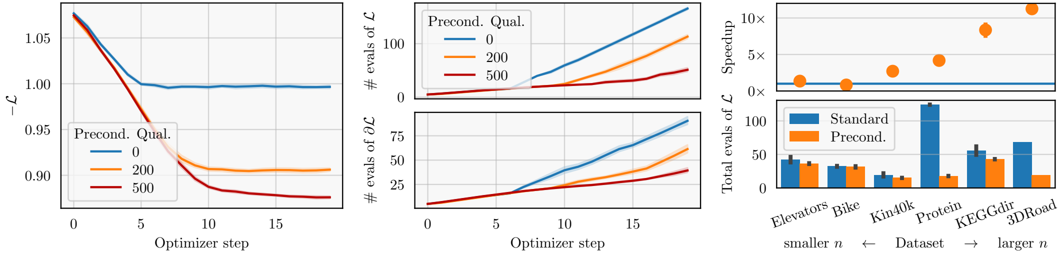


Experiment Details:

- Randomly sampled synthetic data ($n = 10{,}000$, $d = 1$)
- RBF kernel with noise scale $\sigma^2 = 10^{-2}$
- Partial Cholesky preconditioner of size ℓ
- ℓ random vectors

Preconditioning Accelerates Hyperparameter Optimization

Gaussian process hyperparameter optimization on UCI data.



(a) Training loss (Protein).

(b) Line search computations (Protein).

(c) Speedup on UCI datasets.

Experiment Details:

- UCI datasets ($n = 12{,}449$ to $n = 326{,}155$)
- Matérn($\frac{3}{2}$) kernel with noise scale $\sigma^2 = 10^{-2}$
- Partial Cholesky preconditioner of size 500
- $\ell = 50$ random vectors

Summary

Preconditioning for Scalable Gaussian Process Hyperparameter Optimization

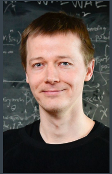
Jonathan Wenger, Geoff Pleiss, Philipp Hennig, John Cunningham and Jacob R. Gardner

- *Preconditioning reduces variance* – or equivalently accelerates convergence – of the stochastic estimates of the $\log$-determinant and its derivatives.
- *Stronger theoretical guarantees* for the computation of the $\log$-determinant, $\log$-marginal likelihood and their derivatives.
- *Specific convergence rates* for combinations of kernels and preconditioners.
- Up to *twelvefold speedup* when training large-scale GP regression models.

Paper arXiv https://arxiv.org/abs/2107.00243

Implementation https://github.com/cornellius-gp/gpytorch





References I

[1] Iain Murray. Gaussian processes and fast matrix-vector multiplies. In Numerical Mathematics in Machine Learning Workshop (ICML), 2009.

[2] Mihai Anitescu, Jie Chen, and Lei Wang. A matrix-free approach for solving the parametric Gaussian process maximum likelihood problem. SIAM Journal on Scientific Computing, 34(1):A240–A262, 2012.

[3] Kurt Cutajar, Michael Osborne, John Cunningham, and Maurizio Filippone. Preconditioning kernel matrices. In International Conference on Machine Learning (ICML), 2016.

[4] Shashanka Ubaru, Jie Chen, and Yousef Saad. Fast estimation of $\mathrm{tr}(f(A))$ via stochastic Lanczos quadrature. SIAM Journal on Matrix Analysis and Applications, 38(4):1075–1099, 2017.

[5] Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. Advances in Neural Information Processing Systems (NeurIPS), 2018:7576–7586, 2018.

[6] Ke Alexander Wang, Geoff Pleiss, Jacob R Gardner, Stephen Tyree, Kilian Q Weinberger, and Andrew Gordon Wilson. Exact Gaussian processes on a million data points. Advances in Neural Information Processing Systems (NeurIPS), 32, 2019.

[7] Artem Artemev, David R Burt, and Mark van der Wilk. Tighter bounds on the log marginal likelihood of Gaussian process regression using conjugate gradients. In International Conference on Machine Learning (ICML), 2021.

[8] Magnus Rudolph Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards, 49, 1952.

[9] Benjamin Charlier, Jean Feydy, Joan Alexis Glaunès, François-David Collin, and Ghislain Durif. Kernel operations on the GPU, with autodiff, without memory overflows. Journal of Machine Learning Research (JMLR), 22(74):1–6, 2021. URL `http://jmlr.org/papers/v22/20-275.html`.

[10] Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. Communications in Statistics-Simulation and Computation, 18(3):1059–1076, 1989.

[11] Gene H Golub and Gérard Meurant. Matrices, moments and quadrature with applications, volume 30. Princeton University Press, 2009.