Contributed Discussion

Simone Rossi*, Cristian Rusu†, Lorenzo A. Rosasco‡, and Maurizio Filippone*

We would like to congratulate with the Authors for this interesting development of probabilistic numerical methods applied to the ubiquitous problem of solving linear systems. We structured this discussion around two main points, namely the use of Bayesian Conjugate Gradient (BCG) for Gaussian processes (GPs), and the possibility to accelerate the solution of linear systems thanks to parallelization of BCG.

Bayesian Conjugate Gradient for Gaussian Processes

Consider a regression task where X and $\mathbf{y}$ denote the set of input points and the set of targets, respectively, and assume a GP with an RBF kernel to model the mapping between X and $\mathbf{y}$ (Rasmussen and Williams, 2006). We are going to assume that GP hyper-parameters are optimized through standard marginal likelihood optimization, although it is possible to reformulate the problem of optimizing GP hyper-parameters in terms of linear systems (Filippone and Engler, 2015) where BCG could be applied. We are going to focus on the predictive distribution and the additional uncertainty stemming from the use of BCG. The GP predictive distribution is $p(\tilde{\mathbf{y}}|X,\mathbf{y},\tilde{X},\boldsymbol{\alpha}) = \mathcal{N}\left(K_{\tilde{X}X}\boldsymbol{\alpha}, \Sigma_{\tilde{\mathbf{y}}}\right)$, where $\boldsymbol{\alpha}$ is the solution of the linear system $(K+\lambda I)\boldsymbol{\alpha} = \mathbf{y}$. As BCG provides a distribution over the solutions for $\boldsymbol{\alpha}$ (i.e. $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\alpha}_m, \Sigma_m)$), we can integrate out $p(\boldsymbol{\alpha})$ obtaining

$$p(\tilde{\mathbf{y}}|X,\mathbf{y},\tilde{X}) = \mathcal{N}\left(K_{\tilde{X}X}\boldsymbol{\alpha}_m, \Sigma_{\tilde{\mathbf{y}}} + K_{\tilde{X}X}\Sigma_m K_{\tilde{X}X}^{\mathrm{T}}\right) ,$$

The topic of preconditioning for solving linear systems involving kernel matrices is an active area of research (Cutajar et al., 2016; Rudi et al., 2017), so we can leverage this in BCG given the connections established in the paper between Σ_0 and preconditioners.

We report the test MNLL and the test RMSE (20% of held-out data) as a function of BCG iterations for two datasets. Figure 1 shows that better preconditioners yield faster convergence. Figure 2 shows the error metrics as a function of time for GPs using BCG and sparse GPs (Matthews et al., 2017). There are configurations where BCG allows to reach better performance for a given computational budget, so this is an interesting possible application of this method.

Bayesian Model Averaging for multiple BCG solutions

One of the advantages that we see in the Bayesian formulation of conjugate gradient, is the possibility to speedup convergence through parallelization. To test this, we solve

*Department of Data Science, EURECOM, Sophia Antipolis, France, simone.rossi@eurecom.fr
†University of Genoa, LCLS – IIT, cristian.rusu@iit.it
‡University of Genoa, LCLS – IIT & MIT, lrosasco@mit.edu
*Department of Data Science, EURECOM, Sophia Antipolis, France, maurizio.filippone@eurecom.fr

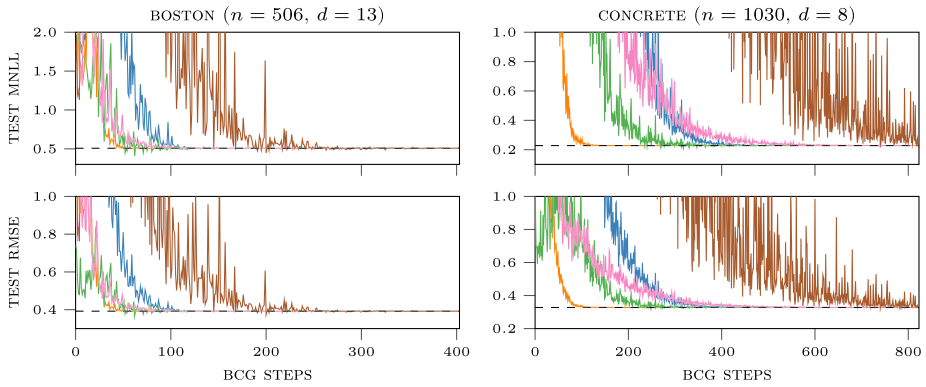

Figure 1: Comparison test MNLL and test RMSE for different priors (e.g. preconditioners) of BCG on two regression datasets. As preconditioners we consider Nyström (Williams and Seeger, 2000) with $\sqrt{n}$ (●) and $4\sqrt{n}$ (●) inducing points, PITC (Candela and Rasmussen, 2005) (●), and RANDOM SVD (Halko et al., 2011) (●). Experiment repeated 25 times.

multiple linear systems with BCG using different priors (possibly concurrently) and aggregate the solutions by means of Bayesian model averaging. Formally, let $\Sigma_0^{(i)}$ denote one of such multiple priors (corresponding to preconditioners) and let $p(\mathbf{x}_m|\Sigma_0^{(i)})$ be the solution at iteration m corresponding to the choice of the ith prior. Assuming a prior on the set of all $\Sigma_0^{(i)}$, the marginalization yields the mixture $p(\mathbf{x}_m) = \sum_i p(\mathbf{x}_m|\Sigma_0^{(i)})p(\Sigma_0^{(i)})$ We project this back to a Gaussian distribution on $p(\mathbf{x}_m)$ by moment matching. We assume a uniform prior for $p(\Sigma_0^{(i)})$, but we could think of relaxing this by setting a prior proportional to the complexity of (or time spent in) inverting the preconditioner.

In Figure 1, the line (●) shows this result. Using the same setup as before, we infer the posterior distribution of a GP using a Bayesian averaging of 16 independent

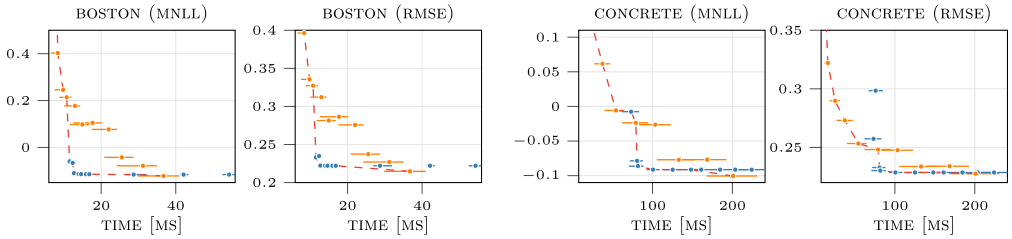

Figure 2: Analysis of the Pareto front (- -) of inference time vs error metric for full GP with BCG (● – Nyström preconditioner is assumed to be precomputed) and sparse GP (●). Points corresponds to different amount of BCG iterations and number of inducing points (with their kernel parameters optimized). Experiment repeated 500 times.

solutions with $\sqrt{n}$ random centers for the Nyström preconditioner (the comparison is with ●). This suggests that it is possible to benefit from combining multiple intermediate solutions of BCG, and this is rather intuitive in the context of Bayesian model averaging.

References

Candela, J. Q. and Rasmussen, C. E. (2005). "A Unifying View of Sparse Approximate Gaussian Process Regression." *Journal of Machine Learning Research*, 6: 1939–1959. MR2249877. 995

Cutajar, K., Osborne, M., Cunningham, J., and Filippone, M. (2016). "Preconditioning Kernel Matrices." In Balcan, M.-F. and Weinberger, K. Q. (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19–24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, 2529–2538. JMLR.org. 994

Filippone, M. and Engler, R. (2015). "Enabling scalable stochastic gradient-based inference for Gaussian processes by employing the Unbiased LInear System SolvEr (ULISSE)." In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, July 6–11, 2015*. 994

Halko, N., Martinsson, P. G., and Tropp, J. A. (2011). "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions." *SIAM Rev.*, 53(2): 217–288. MR2806637. doi: https://doi.org/10.1137/090771806. 995

Matthews, A. G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J. (2017). "GPflow: A Gaussian process library using TensorFlow." *Journal of Machine Learning Research*, 18(40): 1–6. MR3646635. 994

Rasmussen, C. E. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. MIT Press. MR2514435. 994

Rudi, A., Carratino, L., and Rosasco, L. (2017). "FALKON: An Optimal Large Scale Kernel Method." In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, 3888–3898. Curran Associates, Inc. 994

Williams, C. K. I. and Seeger, M. (2000). "Using the Nyström Method to Speed Up Kernel Machines." In Leen, T. K., Dietterich, T. G., Tresp, V., Leen, T. K., Dietterich, T. G., and Tresp, V. (eds.), *NIPS*, 682–688. MIT Press. 995