RANDOMIZED LOW-RANK APPROXIMATION OF KERNEL MATRICES IN GAUSSIAN PROCESSES

ROSS B. ALEXANDER

EE 270 | LARGE SCALE MATRIX COMPUTATION, OPTIMIZATION, AND LEARNING | STANFORD UNIVERSITY

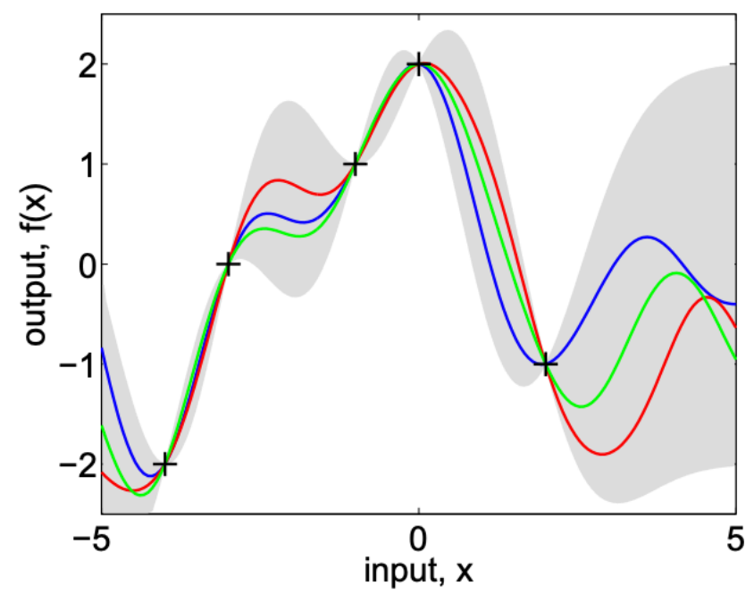
PROBLEM & MOTIVATION

Predictive models in classification & regression tasks
In classification and regression tasks, we propose to fit a particular predictive model to a dataset. These predictive models can range from simple linear models like linear classifiers and regressors, to complex nonlinear models, like deep neural networks.

Parametric models, their success, & their cost
In particular, we are often interested in proposing an expressive, flexible, parametric predictive model that can be fit well to the data given the amount of the data and the model's representational power. While there has recently been immense success in fitting complicated parametric models like deep neural networks, the parameters of the model must be learned through a computationally-intensive training process.

Nonparametric models as an alternative
Nonparametric models offer an alternative to parameter and computation-heavy parametric models. This class of model includes techniques like kernel regression, Dirichlet process mixtures, and other infinite statistical models.

Gaussian processes (GPs)
Gaussian *processes* (GPs) are a class of non-parametric models that are distributions over functions. GPs' parameter complexity scales with the size of the dataset. Moreover, due to their distributional nature, GPs provide direct insight into model uncertainty that deterministic models cannot provide.



A GP (mean and covariance) and sample functions drawn from the GP posterior distribution

Challenges of GPs
One of the challenges GPs have faced is poor sample complexity, since fitting a GP to n samples requires inversion of a kernel matrix which requires operations cubic in the number of samples ($\mathcal{O}(n^3)$). Significant prior work has focused on methods for alleviating this, including subset methods, mixture-of-experts methods, inducing point methods (sparse GPs), and variational approximation [1] [2].

Randomized algorithms for reducing complexity
We address the complexity of the kernel matrix and propose to consider randomized algorithms that can lead to increased parameter efficiency. In particular, we propose to apply the Nyström method [3] for generating consistent low-rank approximations of the kernel matrix. We see applications in tasks involving large datasets (surrogate optimization, active learning, etc.).

METHODS & BACKGROUND

Gaussian processes
We specialize our analysis to 1D GPs without gradient information. We include noise and (w.l.o.g.) use a zero-mean function *m*, and a squared exponential kernel function *k*. Both the noise σ, and length scale ℓ, are not learned in the process. We can compute the predictive distribution mean and variance, *μ-hat* and *ν-hat*.

$$\mathcal{D} = \{X, y\} \qquad X \in \mathbb{R}^{n\times d}, y \in \mathbb{R}^n$$

$$\hat{\mu}(x) = m(x) + K(x,X)(K(X,X)+\sigma I)^{-1}(y - m(x))$$

$$\hat{\nu}(x) = K(x,x) + K(x,X)(K(X,X)+\sigma I)^{-1}K(X,x)$$

Nyström method
Generate a column subsampling matrix S (without replacement) using a sampling method (uniform, ℓ_2, inverse ℓ_2, etc.).

$$S \in \mathbb{R}^{n\times p}$$

Generate a low-rank approximation of the kernel matrix & invert quickly using matrix inversion lemma.

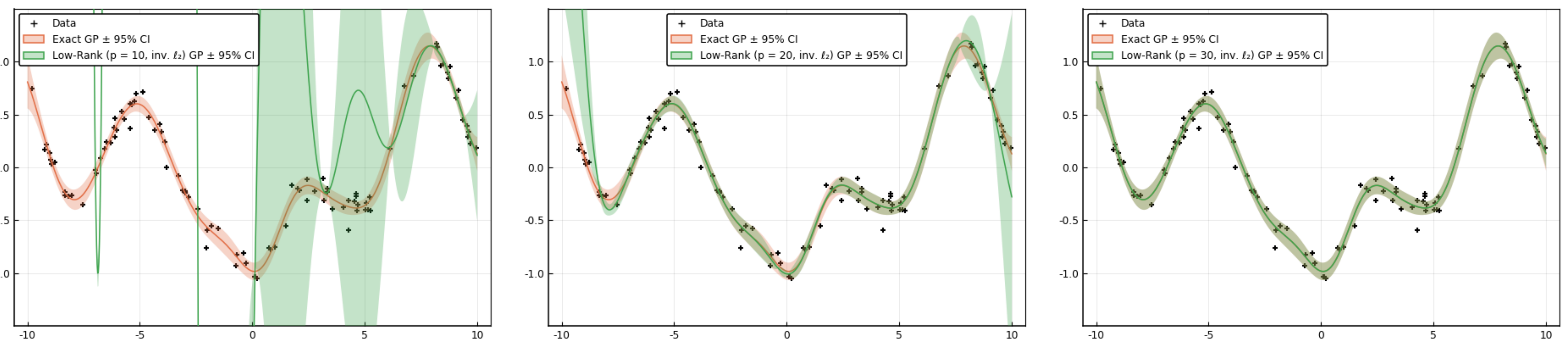
$$K = U\Lambda U^\top \qquad S^\top KS = U_p\Lambda_p U_p^\top$$

$$\tilde{K} = \tilde{U}\tilde{\Lambda}\tilde{U}^\top \qquad \tilde{\lambda}_i^{(n)} \approx \frac{n}{p}\lambda_i^{(p)}$$

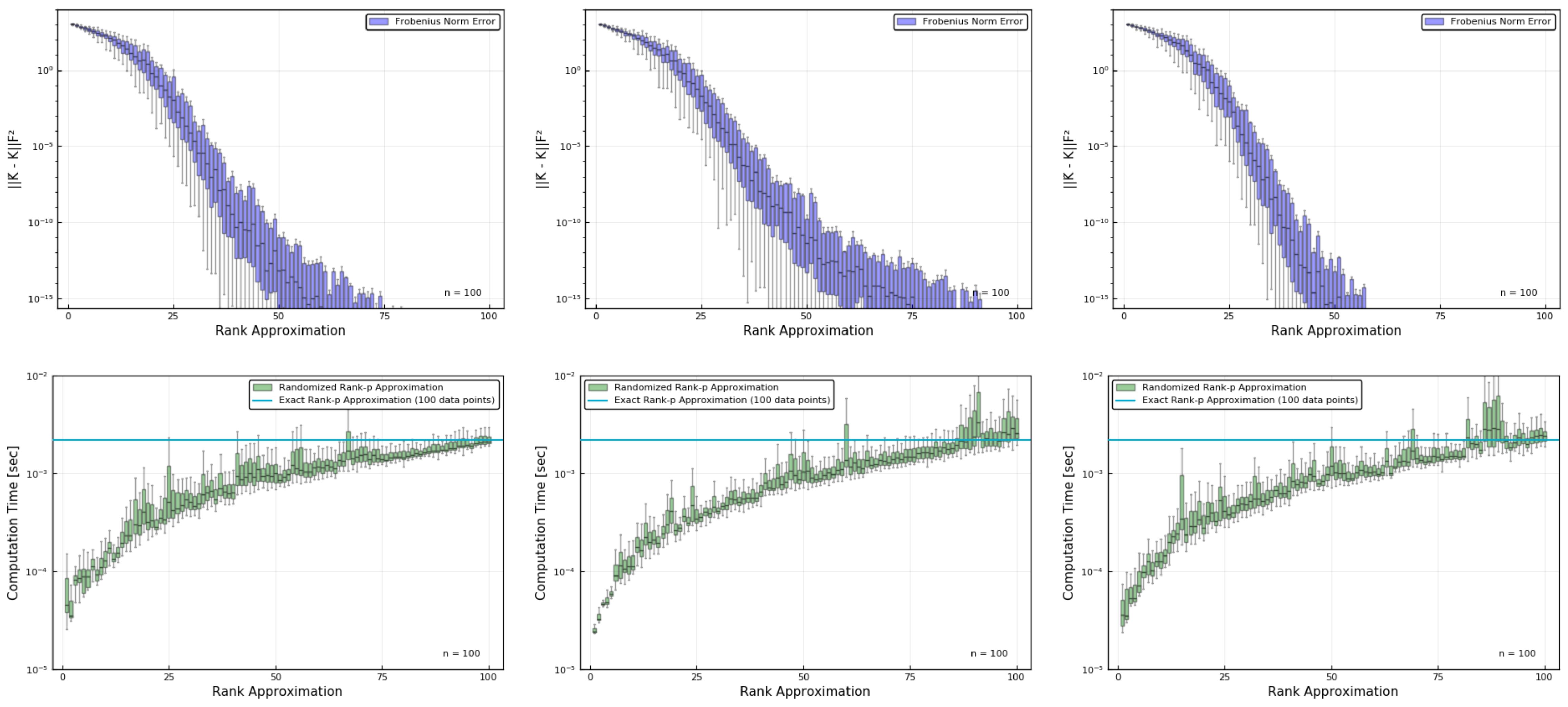
$$\tilde{K} = \sum_{i=1}^{p} \tilde{\lambda}_i^{(n)}\tilde{u}_i^{(n)}\tilde{u}_i^{(n)\top} \qquad \tilde{u}_i^{(n)} \approx \sqrt{\frac{p}{n}}\frac{1}{\lambda_i^{(n)}}KSu_i^{(p)}$$

RESULTS

Full-Rank GP and Randomized Low-Rank GP using Inverse ℓ_2 Score Column Sampling


p = 10
p = 20
p = 30

Variation in Kernel Matrix Approximation Error and Computation Time by Approximation Rank


Uniform Column Subsampling
ℓ_2 Score Column Subsampling
Inverse ℓ_2 Score Column Subsampling

$$f(x) = \tfrac{1}{3}\log\left(1 + |x|^{2+\sin x}\right) - 1 + \tfrac{u}{12} \qquad x \sim \mathcal{U}(-10, 10) \qquad u \sim \mathcal{N}(0,1)$$

*Results available for uniformly and non-uniformly distributed samples.

**Boxplots for each value of approximation rank *p* are aggregated over 100 random runs.

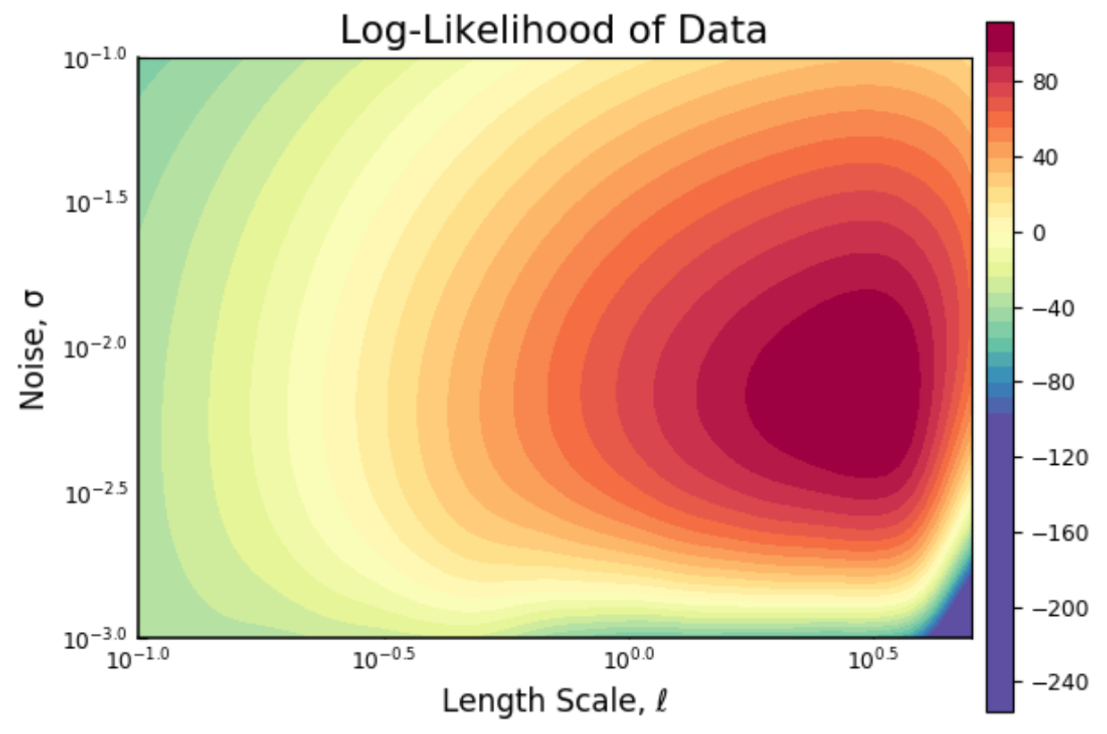
DISCUSSION

Low-rank approximations lead to feasible GPs
For the function we tested, in both the uniformly- and non-uniformly-sampled data cases with 100 data points, we find suitable GPs even for approximation rank as low as *p*=20. As the approximation rank approaches half of the number of data points, the randomized low-rank approximation of the kernel matrix is essentially identical to the original kernel matrix.

Inverse ℓ_2 score column subsampling performs best
We find that (perhaps unsurprisingly) upsampling rows that have low ℓ_2-norm (inverse ℓ_2 score column subsampling) leads to superior low-rank approximations. In particular, the insight is that we upsample rows of the kernel matrix corresponding to samples that are far away from other samples in the dataset, which improves the performance of the GP over the entire domain.

Low-rank approximations are considerably faster
In view of the computation time, we see that generating the randomized projection matrix and constructing the randomized low-rank approximation of the kernel matrix is faster than performing an exact low-rank approximation when the approximation rank is less than the number of samples.

FUTURE WORK



There are lots of avenues for future work, including exploring additional, higher-dimensional functions with gradient information. We also would like to investigate the kernel learning process as well as low-rank deep GPs.

REFERENCES

[1] Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai. When gaussian process meets big data: A review of scalable gps, 2019.

[2] Carl Edward Rasmussen. Gaussian processes in machine learning. In Summer school on machine learning, pages 63–71. Springer, 2003.

[3] Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In T. Leen, T. Dietterich, and V. Tresp, editors, Advances in Neural Information Processing Systems, volume 13. MIT Press, 2001.