Randomized Low-Rank Approximation of Kernel Matrices in Gaussian Processes

Ross Alexander
Department of Aeronautics & Astronautics
Stanford University
rbalexan@stanford.edu

1 Introduction

In traditional 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. We are often interested in proposing a parametric predictive model that balances intrinsic representational capacity with the algorithmic and computational effort needed to fit the model parameters to the dataset. While there has recently been immense success in fitting representationally-complex parametric models like deep neural networks (DNNs) to large datasets, the parameters of these DNNs must be learned through a computationally-intensive training process.

Non-parametric models offer an alternative to both parameter- and computation-heavy parametric models. There are a variety of non-parametric models including techniques like kernel classifiers, kernel regressors, Dirichlet process mixture models, and other infinite statistical models.

Gaussian processes (GPs) are a class of non-parametric models that represent distributions over functions. Like all non-parametric models, GPs' parameter complexity scales with the size of the dataset. As stochastic models, GPs can provide direct insight into model uncertainty that deterministic models cannot provide. This model uncertainty is often especially important in real-world systems, where strong guarantees of model performance are critical.

However, while Gaussian processes are powerful parameter-efficient models, they face significant challenges in sample complexity. Fitting a GP to n samples requires storing $\mathcal{O}(n^2)$ and inverting $\mathcal{O}(n^3)$ a kernel matrix. Due to the cubic sample complexity of kernel matrix inversion, GPs are not widely used for problems involving large datasets. Reducing the sample complexity of GPs has been the a significant focus in the last two decades and much progress has been made. In particular, a variety of approaches have been centered around identifying or constructing sufficient subsets of the points to use in forming the kernel matrix. Nominally, these subset methods include a class of methods called inducing point methods and so-called sparse GPs. There has been additional research in mixture-of-experts methods for fitting multiple smaller GPs over subsets of the problem domain; other work on variational approximation has also been quite successful [2, 1].

In this project, we consider randomized subsampling algorithms for generating low-rank approximations of the kernel matrix. In particular, we generate a sketching matrix by subsampling columns of the kernel matrix without replacement according to a specified distribution. We investigate three sampling distributions: a uniform column distribution, an ℓ_2-norm-square column score distribution, and an inverse ℓ_2-norm-square column score distribution. We then use the Nyström method [3] on our kernel matrix and sketching matrix to construct consistent low-rank approximations of the kernel matrix through approximate eigendecompositions. Finally, we apply the matrix inversion lemma to quickly invert the low-rank kernel matrix. We apply our approach to a function with samples spread uniformly and non-uniformly over the problem domain. Our results show that low-rank approximations can be achieved in times linear with the approximation rank, and therefore, in times sub-cubic with the number of samples. Using a squared Frobenius norm metric, the low-rank kernel matrix

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

approximations converge quickly to the full-rank kernel matrix and the inverse ℓ_2-norm-squared column score distribution provides the fastest convergence among the sampling distributions tested.

In Section 2, we introduce our notation and review the relevant theory behind Gaussian processes. Section 3 provides descriptions of the sampling distributions and sketching matrices and also covers our application the Nyström method and the matrix inversion lemma. Our experiments and results are presented and discussed in Section 4. We summarize the main conclusion in Section 5.

2 Background

Suppose we are given a dataset $\mathcal{D} = \{x^{(i)}, y^{(i)}\}_{i=1}^{n}$ consisting of inputs $x^{(i)} \in \mathbb{R}^n$ and outputs $y^{(i)} \in \mathbb{R}$. [1] We assume that the outputs are noisy measurements from an underlying process f, such that $y^{(i)} = f(x^{(i)}) + \epsilon^{(i)}$ with $\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$ for some arbitrary σ. We write the dataset concisely by denoting the input as a data matrix $X \in \mathbb{R}^{n \times d}$ and denoting the output as a target vector $y \in \mathbb{R}^n$.

We introduce the Gaussian process (GP), denoted $\mathcal{GP}(m, k)$, where $m : \mathbb{R}^d \to \mathbb{R}$ is the mean function and $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the (symmetric) kernel function. Since a GP is a joint distribution over outputs given inputs, a GP is inherently a distribution over functions. For a GP over dataset $\mathcal{D}$, we have:

$$\begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x^{(1)}) \\ \vdots \\ m(x^{(n)}) \end{bmatrix}, \begin{bmatrix} k(x^{(1)}, x^{(1)}) & \cdots & k(x^{(1)}, x^{(n)}) \\ \vdots & \ddots & \vdots \\ k(x^{(n)}, x^{(1)}) & \cdots & k(x^{(n)}, x^{(n)}) \end{bmatrix} \right). \tag{1}$$

In a more concise form, we can write:

$$y \sim \mathcal{N}\left(m(X), K(X, X)\right) \tag{2}$$

where we have the general mean vector function $m : \mathbb{R}^{n \times d} \to \mathbb{R}^n$ and (symmetric) kernel matrix function $K : \mathbb{R}^{n \times d} \times \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times n}$.

$$m(X) = \begin{bmatrix} m(x^{(1)}) \\ \vdots \\ m(x^{(n)}) \end{bmatrix} \tag{3}$$

$$K(X, X') = \begin{bmatrix} k(x^{(1)}, x'^{(1)}) & \cdots & k(x^{(1)}, x'^{(n')}) \\ \vdots & \ddots & \vdots \\ k(x^{(n)}, x'^{(1)}) & \cdots & k(x^{(n)}, x'^{(n')}) \end{bmatrix} \tag{4}$$

Now, formulating the joint distribution over new outputs $\hat{y} \in \mathbb{R}^{n^*}$ and our outputs y, given new inputs $X^* \in \mathbb{R}^{n^* \times d}$ and our inputs X, we have:

$$\begin{bmatrix} \hat{y} \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(X^*) \\ m(X) \end{bmatrix}, \begin{bmatrix} K(X^*, X^*) & K(X^*, X) \\ K(X, X^*) & K(X, X) \end{bmatrix} \right). \tag{5}$$

Since Gaussian distributions are closed under marginalization, the posterior (predictive) distribution $p(\hat{y} \mid y)$ can be computed analytically:

$$\hat{y} \mid y \sim \mathcal{N}(m(X^*) + K(X^*, X)K(X, X)^{-1}(y - m(X)), \tag{6}$$

$$K(X^*, X^*) - K(X^*, X)K(X, X)^{-1}K(X, X^*)). \tag{7}$$

As is common practice, we include a noise parameter σ^2 (analogous to weak regularization) to admit noise into the Gaussian process and to improve numerical stability of the required kernel matrix inversion. After the addition of this term, we separate out the predicted mean function $\hat{\mu} : \mathbb{R}^n \to \mathbb{R}$ and the predicted variance function $\hat{\nu} : \mathbb{R}^n \to \mathbb{R}^n$:

$$\hat{\mu}(x) = m(x) + K(x, X)(K(X, X) + \sigma^2 I)^{-1}(y - m(x)) \tag{8}$$

$$\hat{\nu}(x) = K(x, x) + K(x, X)(K(X, X) + \sigma^2 I)^{-1}K(X, x) \tag{9}$$

[1] Our results generalize for multidimensional outputs.

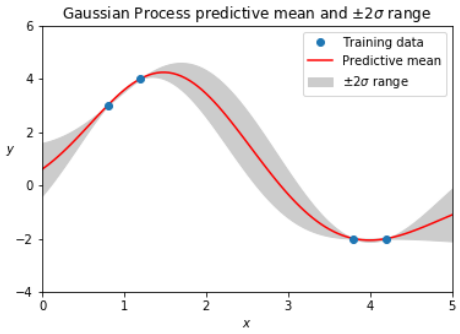

Figure 1: Sample Gaussian process (GP) with dataset (blue), predicted mean function $\hat{\mu}$ (red), and predicted $\pm 2\hat{\sigma}$ confidence interval $\left(\pm 2\sqrt{\hat{\nu}(x)}\right)$.

We can observe a sample Gaussian process in Figure 1, with mean and variance indicated. We can also sample functions from the posterior distribution by sequentially sampling a series of outputs over the domain.

3 Methodology

We investigate three different sketching methodsfor subsampling the kernel matrix: (1) uniform column sketching, (2) ℓ_2-score column sketching, and (3) inverse ℓ_2-score column sketching. We form the sketch matrix $S \in \mathbb{R}^{n \times p}$ by concatenating p randomly-sampled columns *without replacement* according to the chosen sampling distribution. Specifically, we sample a column index $i_t \in 1, \cdots, n$ from the distribution $p(i_t)$ and concatenate column $K^{(i_t)}$ with S. We have the following three distributions:

$$\begin{aligned} \text{UNIFORM}: &\quad p(i_t) = \tfrac{1}{n} \\ \text{L2SCORE}: &\quad p(i_t) \propto ||K^{(i_t)}||_2^2 \\ \text{INVERSEL2SCORE}: &\quad p(i_t) \propto \tfrac{1}{||K^{(i_t)}||_2^2} \end{aligned}$$

After sampling a matrix S we can construct left-, right-, and symmetrically-sketched versions of the kernel matrix: $S^\top K$, KS, and $S^\top KS$, respectively. We can apply the Nystrom method to construct the low-rank approximation of the kernel matrix $\tilde{K}$ as follows:

$$\tilde{K} = KS(S^\top KS)^{-1} S^\top K \in \mathbb{R}^{n \times n} \tag{10}$$

However, we would like the eigendecomposition of the low-rank kernel matrix approximation so that we can quickly invert it. Both the kernel matrix and its low-rank approximation are symmetric and positive definite, i.e. $K, \tilde{K} \in \mathbb{S}_{++}^n$. As a result of symmetry, they admit eigendecompositions $K = U\Lambda U^\top$ and $\tilde{K} = \tilde{U}\tilde{\Lambda}\tilde{U}^\top$ where the respective Λ matrices are diagonal matrices of the sorted eigenvalues ($\Lambda = \text{diag}(\lambda_{[1]}, ..., \lambda_{[n]})$ where $\lambda_{[i]} \geq \lambda_{[j]} \forall i > j \in \{1, \cdots, n\}$) and where the respective U matrices are orthogonal matrices whose columns are the paired eigenvectors. Our low-rank (p-rank, to be precise) kernel matrix approximation can be written as:

$$\tilde{K} = \tilde{U}\tilde{\Lambda}\tilde{U}^\top \tag{11}$$

$$\tilde{K} = \sum_{i=1}^{p} \tilde{\lambda}_i^{(n)} \tilde{u}_i^{(n)} \tilde{u}_i^{(n)\top} \tag{12}$$

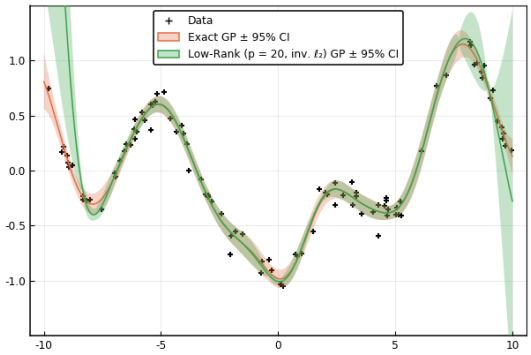

Figure 2: Exact GP (red) and low-rank ($p = 20$) GP approximated using INVERSEL2SCORE sampling (green) for 100 non-uniformly spread inputs.

Using an eigendecomposition of the symmetric sketch of the kernel matrix, we can approximate the eigendecomposition of the full-rank kernel matrix.

$$S^\top K S = U_p \Lambda_p U_p^\top \tag{13}$$

$$\tilde{\Lambda} \approx \frac{n}{p} \Lambda_p \tag{14}$$

$$\tilde{U} \approx \sqrt{\frac{p}{n}} K S U_p \Lambda_p^{-1} \tag{15}$$

Finally, we can quickly invert the low-rank kernel matrix approximation (plus thus noise-identity term) using the matrix inversion lemma, i.e.

$$(\tilde{K} + \sigma^2 I)^{-1} = \tfrac{1}{\sigma^2}(I - \tilde{U}(\sigma^2 I + \tilde{\Lambda}\tilde{U}^\top \tilde{U})^{-1}\tilde{\Lambda}\tilde{U}^\top) \tag{16}$$

This ultimately reduces the cubic complexity of the inversion task ($\mathcal{O}(n^3)$) to the complexity of the sketch generation, sketched eigendecomposition, and inversion task, which is $\mathcal{O}(pn^2)$.

4 Experiments Results & Discussion

For our experiments, we consider a zero-mean function and squared-exponential (Gaussian) kernel:

$$m(x) = 0 \tag{17}$$

$$k(x, x') = \exp\left(-\frac{1}{2}||x - x'||_2^2\right). \tag{18}$$

We also set our noise parameter $\sigma^2 = 1 \times 10^{-2}$ and our kernel length scale $\ell = 2$ (these parameters can be tuned or learned).

We generate data for a 1D function in two regimes: (1) uniformly distributed over the domain and (2) non-uniformly distributed over the domain. We choose a function $f(x) = \frac{1}{3}\log\left(1 + |x|^{2+\sin x}\right) - 1 + \frac{1}{12}\varepsilon$ where $\varepsilon \sim \mathcal{N}(0, 1)$. For the dataset, we fit the exact GP and the low-rank GP, depicted in Figure 2. For each sampling method, we perform 100 trials at each approximation rank p for $p \in \{1, \cdots, 100\}$. In each of these trials, we compute the squared Frobenius norm approximation error $||\tilde{K} - K||_F^2$ and we track the computation time of generating and inverting the low-rank approximation of the kernel matrix.

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.

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

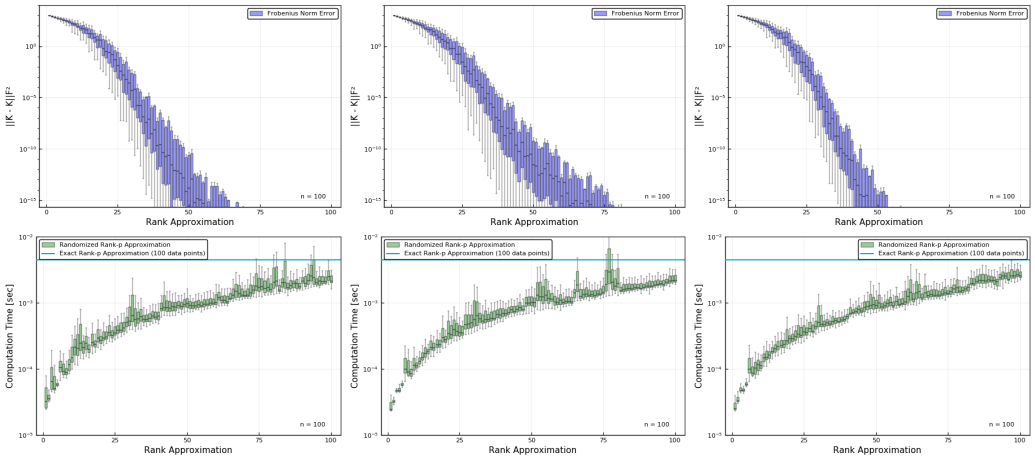

Figure 3: Results for ***uniformly distributed*** input data for UNIFORM (left), L2SCORE (middle), and INVERSEL2SCORE (right) sampling methods. Upper figures depict variation of squared Frobenius norm error of the randomized low-rank kernel matrix approximation with the approximation rank p. Lower figures depict variation in computation time with the approximation rank p. Each boxplot in the each figure represents an outlier-removed aggregation of 100 random runs.

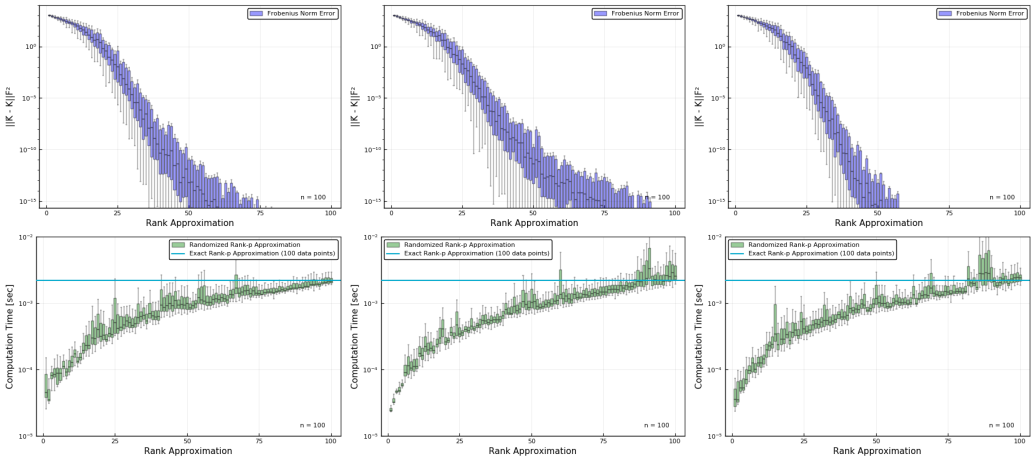

Figure 4: Results for ***non-uniformly distributed*** input data for UNIFORM (left), L2SCORE (middle), and INVERSEL2SCORE (right) sampling methods. Upper figures depict variation of squared Frobenius norm error of the randomized low-rank kernel matrix approximation with the approximation rank p. Lower figures depict variation in computation time with the approximation rank p. Each boxplot in the each figure represents an outlier-removed aggregation of 100 random runs.

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.

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.

5 Conclusion

In this project, we considered randomized subsampling algorithms for generating low-rank approximations of the kernel matrix. We generated sketching matrices by subsampling columns of the kernel matrix without replacement according to three specified distributions. We then leveraged the Nyström method [3] on our kernel matrix and sketching matrix to construct consistent low-rank approximations of the kernel matrix through approximate eigendecompositions. Finally, we applied the matrix inversion lemma to quickly invert the low-rank kernel matrix. We applied our approach to a function with samples spread uniformly and non-uniformly over the problem domain. Our results show that low-rank approximations can be achieved in times linear with the approximation rank, and therefore, in times sub-cubic with the number of samples. Using a squared Frobenius norm metric, the low-rank kernel matrix approximations converge quickly to the full-rank kernel matrix and the inverse ℓ_2-norm-squared column score distribution provide the fastest convergence among the sampling distributions tested. We see many applications iof our method in tasks involving large datasets (surrogate optimization, active learning, etc.).

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.