Kernel Distillation for Fast Gaussian Processes
Prediction

Congzheng Song∗
Cornell Tech
cs2296@cornell.edu

Yiming Sun∗
Cornell University
ys784@cornell.edu

Abstract

Gaussian processes (GPs) are flexible models that can capture complex structure in large-scale dataset due to their non-parametric nature. However, the usage of GPs in real-world application is limited due to their high computational cost at inference time. In this paper, we introduce a new framework, kernel distillation, to approximate a fully trained teacher GP model with kernel matrix of size $n \times n$ for $n$ training points. We combine inducing points method with sparse low-rank approximation in the distillation procedure. The distilled student GP model only costs $O(m^2)$ storage for $m$ inducing points where $m \ll n$ and improves the inference time complexity. We demonstrate empirically that kernel distillation provides better trade-off between the prediction time and the test performance compared to the alternatives.

1 Introduction

Gaussian Processes (GPs) [1] are powerful tools for regression and classification problems as these models are able to learn complex representation of data through expressive covariance kernels. However, the application of GPs in real-world is limited due to their poor scalability during inference time. For a training data of size $n$, GPs requires $O(n^2)$ computation and storage for inference a single test point. Previous way for scaling GP inference is either through inducing points methods [2, 3, 4] or structure exploitation [5, 6]. More recently, the structured kernel interpolation (SKI) framework and KISS-GP [7] further improve the scalability of GPs by unifying inducing points methods and structure exploitation. These methods can suffer from degradation of test performance or require input data to have special grid structure.

All the previous solutions for scaling GPs focus on training GPs from scratch. In this paper, we focus on a different setting where we have enough resource for training exact GPs and want to apply the trained model for inference on resource-limited devices such as mobile phone or robotics [8]. We wish to investigate the possibility of compressing a large trained exact GP model while preserve the predictive power of the exact model. This paper proposes kernel distillation, a general framework to approximate a trained GP model. Kernel distillation extends inducing point methods with insights from SKI framework and utilizes the knowledges from a trained model.

In particular, we approximate the exact kernel matrix with a sparse and low-rank structured matrix. We formulate kernel distillation as a constrained $F$-norm minimization problem, leading to more accurate kernel approximation compared to previous approximation approaches. Our method is a general purpose kernel approximation method and does not require kernel function to be separable or stationary and input data to have any special structure. We evaluate our approach on various real-world datasets, and the empirical results evidence that kernel distillation can better preserving the predictive power of a fully trained GP model and improving the speed simultaneously compared to other alternatives.

∗Equal contribution
2 Kernel Distillation

Background. We focus on GP regression problem. Denote the dataset as \( D \) which consists of input feature vectors \( X = \{x_1, \ldots, x_n\} \) and real-value targets \( y = \{y_1, \ldots, y_n\} \). GP models a distribution over functions \( f(x) \sim GP(\mu, \kappa) \), where any set of function values forms a joint Gaussian distribution characterized by mean function \( \mu(x) \) and kernel function \( \kappa(x, \cdot) \) where \( \gamma \) is the set of hyper-parameters to be trained. Based on Gaussian Identity, we can arrive at posterior predictive distribution for inference \cite{Rasmussen2006}:

\[
f|X, X*, y \sim \mathcal{N}(K_{XX}(K_{XX} + \sigma^2 I)^{-1}y, \quad K_{X*, X} - K_{X*, X}(K_{XX} + \sigma^2 I)^{-1}K_{XX}).
\]

The matrix \( K_{XX} = \kappa(X, X) \) is the covariance measured between \( X* \) and \( X \). The prediction for mean and variance cost \( O(n^3) \) in time and \( O(n^2) \) in storage per test point.

The computational and storage bottleneck is the exact kernel matrix \( K_{XX} \). Kis-GP \cite{Gal2016} is a inducing point method for approximating the kernel matrix and thus scaling training of GPs. Given a set of \( m \) inducing points \( U = \{u_1, \ldots, u_m\} \), Kis-GP approximates the kernel matrix \( K_{XX} \approx W_X K_{UU} W_X^\top \) where \( X \) is locally interpolated with \( U \) and \( W_X \) is the interpolation weights.

Formulation. The goal of kernel distillation is to compress a fully trained GP model to an approximate GP model to be used for inference on a resource-limited devices. We assume that we have access to a trained exact GP with full kernel matrix \( K_{XX} \) and all the training data \( \{X, y\} \) during distillation. Algorithm 1 in Appendix A outlines our distillation procedure.

We propose to use a student kernel matrix with a sparse and low-rank structure, \( \tilde{K}_{XX} = WK_{UU}W^\top \) to approximate a fully trained kernel matrix \( K_{XX} \). \( W \) is a sparse matrix and \( K_{UU} \) is the covariance evaluated at a set of inducing points \( U \). Similar to Kis-GP \cite{Gal2016}, we approximate \( K_{XX} \) with \( K_{UU} = WK_{UU} \). In Kis-GP, \( W \) is calculated using cubic interpolation on grid-structured inducing points. The number of inducing points grows exponentially as the dimension of input data grows, limiting Kis-GP applicable to low-dimensional data. Instead of enforcing inducing points \( U \) to be on grid, we choose \( m \) centroids from the results of K-means clustering \( X \) as the inducing points \( U \). In addition, we store \( U \) in KD-tree \( T \) for fast nearest neighbor search which will be used in later optimization.

In kernel distillation, we find optimal \( W \) through a constrained optimization problem. We constrain each row of \( W \) to have at most \( b \) non-zero entries. We set the objective function to be the \( F \)-norm error between teacher kernel and student kernel:

\[
\min_W \| K_{XX} - WK_{UU}W^\top \|_F
\]

subject to \( \| W_i \|_0 \leq b \quad \forall i \)

where \( \| W_i \|_0 \) denotes the number of non-zero entries at row \( i \) of \( W \).

Initializing \( W \). The initial values of \( W \) are crucial for the later optimization. We initialize \( W \) with optimal solution to \( \| K_{XX} - WK_{UU} \|_F \) with the sparsity constraint. More specifically, for each \( x_i \) in \( X \), we find its \( b \) nearest points in \( U \) by querying \( T \). We denote the indices of these \( b \) neighbors as \( J \). We then initialize each row \( W_i \) of \( W \) by solving the following linear least square problem:

\[
\min_{W_i(J)} \| W_i(J) K_{UU}(J) - (K_{XX})_i \|_2^2
\]

where \( W_i(J) \) denotes the entries in row \( W_i \) indexed by \( J \) and \( K_{UU}(J) \) denotes the rows of \( K_{UU} \) indexed by \( J \). The entries in \( W_i \) with index not in \( J \) are set to zero.

Optimizing \( W \). After \( W \) is initialized, we solve the \( F \)-norm minimization problem using standard gradient descent. To satisfy the sparsity constraint, in each iteration, we project each row of the gradient \( \nabla W \) to \( b \)-sparse space according the indices \( J \), and then update \( W \) accordingly.

Fast prediction. One direct application of kernel distillation is for fast prediction with approximated kernel matrix. Given a test point \( x_* \), we follow similar approximation scheme in the distillation at test time where we try to approximate \( K_{x, x} \):

\[
K_{x, x} \approx \tilde{K}_{x, x} = W_* K_{UU} W_*^\top
\]
where $W_*$ is forced to be sparse for efficiency. Then the mean and variance prediction can be approximated by:

$$E[f_*] \approx \tilde{K}_{x, \mathcal{X}} (\tilde{K}_{XX} + \sigma^2 I)^{-1} y \approx W_* K_{UU} W_*^\top (\tilde{K}_{XX} + \sigma^2 I)^{-1} y$$

$$\text{Var}[f_*] \approx K_{x_*, x} - \tilde{K}_{x_*, \mathcal{X}} [\tilde{K}_{XX} + \sigma^2 I]^{-1} \tilde{K}_{x_*, \mathcal{X}}$$

$$\approx K_{x_*, x} - W_* K_{UU} W_*^\top [\tilde{K}_{XX} + \sigma^2 I]^{-1} W K_{UU} W_*^\top$$

$$= K_{x_*, x} - W_* V W_*^\top$$

where both $\hat{\alpha} = K_{UU} W_*^\top (\tilde{K}_{XX} + \sigma^2 I)^{-1} y$ and $V = K_{UU} W_*^\top [\tilde{K}_{XX} + \sigma^2 I]^{-1} W K_{UU}$ can be precomputed during distillation.

To compute $W_*$ efficiently, we start by finding $b$ nearest neighbors of $x_*$ in $U$ (indexed by $J_*$) and set the entries in $W_*$ whose indices are not in $J_*$ to 0. For entries with indices in $J_*$, we solve the following least square problem to get the optimal values for $W_*(J_*)$:

$$\min_{W_*(J_*)} \| W_*(J_*) K_{UU} (J_*) - K_{x_*, U}(J_*) \|_2.$$ 

It takes $O(b \log m)$ to query the nearest neighbors, $O(b^3)$ to get $W_*$ and $O(b)$ and $O(b^2)$ for mean and variance prediction respectively. The prediction time complexity is $O(b \log m + b^3)$ in total. As for storage complexity, we need to store precomputed vector for mean prediction and diagonal of matrix for variance prediction which cost $O(m^2)$. Table 1 provides comparison of time and storage complexity for different GP approximation approaches.

### 3 Experiments

We evaluate kernel distillation on the ability to approximate the exact kernel, the predictive power and the speed at inference time. In particular, we compare our approach to FITC and KISS-GP as they are the most popular approaches and are closely related to kernel distillation. The simulation experiments for reconstructing kernel and comparing predictive power are demonstrated in Appendix B.

**Empirical Study.** We evaluate the performance of kernel distillation on several benchmark regression data sets. A summary of the datasets is given in Table 2. The detailed setup of experiments is in Appendix C.

We start by evaluating how well kernel distillation can preserve the predictive performance of the teacher kernel. The metrics we use for evaluation is the standardized mean square error (SMSE) defined as $\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 / \text{Var}(y)$ for true label $y_i$ and model prediction $\hat{y}_i$. Table 2 summarizes the results. We can see that exact GPs achieve lowest errors on all of the datasets. FITC gets second lowest error on almost all datasets except for Boston Housing. Errors with kernel distillation are very close to FITC while KISS-GP has the largest errors on every dataset. The poor performance of KISS-GP might be resulted from the loss of information through the projection of input data to low dimension.

We further study the effects of sparsity $b$ on predictive performance. We choose $b$ to be range from [5, 10, …, 40] and compare the test error and variance prediction for KISS-GP and kernel distillation on Boston Housing and Abalone datasets. The results are shown in Figure 1. As expected, the error for kernel distillation decreases as the sparsity increases and we only need $b$ to be 15 or 20 to
Table 2: SMSE Results Comparison. $d$ is the dimension of the input data. Number of inducing points (on 2D grid) for KISS-GP are 4,900, 10K, 90K, 250K, and number of inducing points for FITC and kernel distillation are 70, 200, 1K, 1K for the for datasets respectively. The sparsity $b$ is set to 20 for Boston Housing and 30 for all other datasets.

| Dataset       | $d$ | # train | # test | Exact     | FITC     | KISS-GP  | Distill |
|---------------|-----|---------|--------|-----------|----------|----------|---------|
| Boston Housing| 13  | 455     | 51     | 0.076     | 0.103    | 0.095    | 0.091   |
| Abalone       | 8   | 3,133   | 1,044  | 0.434     | 0.438    | 0.446    | 0.439   |
| PUMADYM32N    | 32  | 7,168   | 1,024  | 0.044     | 0.044    | 1.001    | 0.069   |
| KIN40K        | 8   | 10,000  | 30,000 | 0.013     | 0.030    | 0.386    | 0.173   |

(a) Boston Mean (b) Boston Variance (c) Abalone Mean (d) Abalone Variance

Figure 1: Test error and variance comparison on Boston Housing (a-b) and Abalone (c-d) under different choices of sparsity constraint $b$ on $W$. For variance prediction comparison, we calculate the root square mean error between variance of exact GPs and approximate GPs (KISS-GP and kernel distillation).

outperform KISS-GP. As for variance prediction, we plot the error between outputs from exact GPs and approximate GPs. We can see that kernel distillation always provides more reliable variance output than KISS-GP on every level of sparsity.

Finally, we evaluate the speed of prediction with kernel distillation. Again, we compare the speed with FITC and KISS-GP. The setup for the approximate models is the same as the predictive performance comparison experiment. For each dataset, we run the test prediction on 1000 points and report the average prediction time in seconds. Table 3 summarizes the results on speed. It shows that both KISS-GP and kernel distillation are much faster in prediction time compared to FITC for all datasets. Though kernel distillation is slightly slower than KISS-GP, considering the improvement in accuracy and more reliable uncertainty measurements, the cost in prediction time is acceptable. Also, though KISS-GP claims to have constant prediction time complexity in theory [11], the actual implementation still is data-dependent and the speed varies on different datasets. In general, kernel distillation provides a better trade-off between predictive power and scalability than its alternatives.

Conclusion. We proposed a general framework, kernel distillation, for compressing a trained exact GPs kernel into a student kernel with low-rank and sparse structure. Our framework does not assume any special structure on input data or kernel function, and thus can be applied "out-of-box" on any datasets. Kernel distillation framework formulates the approximation as a constrained $F$-norm minimization between exact teacher kernel and approximate student kernel.

The distilled kernel matrix reduces the storage cost to $O(m^2)$ compared to $O(mn)$ for other inducing point methods. Moreover, we show one application of kernel distillation is for fast and accurate GP prediction. Kernel distillation can produce more accurate results than KISS-GP and the prediction time is much faster than FITC. Overall, our method provide a better balance between speed and predictive performance than other approximate GP approaches.

Table 3: Average prediction time in seconds for 1K test data points

| Dataset       | FITC    | KISS-GP  | Distill |
|---------------|---------|----------|---------|
| Boston Housing| 0.0081  | 0.00061  | 0.0017  |
| Abalone       | 0.0631  | 0.00018  | 0.0020  |
| PUMADYM32N    | 1.3414  | 0.0011   | 0.0035  |
| KIN40K        | 1.7606  | 0.0029   | 0.0034  |
References

[1] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.

[2] Matthias Seeger, Christopher Williams, and Neil Lawrence. Fast forward selection to speed up sparse gaussian process regression. In *Artificial Intelligence and Statistics 9*, number EPFLCONF-161318, 2003.

[3] Michalis K Titsias. Variational learning of inducing variables in sparse gaussian processes. In *AISTATS*, volume 12, pages 567–574, 2009.

[4] Neil Lawrence, Matthias Seeger, and Ralf Herbrich. Fast sparse gaussian process methods: The informative vector machine. In *Proceedings of the 16th Annual Conference on Neural Information Processing Systems*, number EPFL-CONF-161319, pages 609–616, 2003.

[5] Yunus Saatç़ि. *Scalable inference for structured Gaussian process models*. PhD thesis, University of Cambridge, 2012.

[6] Andrew Wilson, Elad Gilboa, John P Cunningham, and Arye Nehorai. Fast kernel learning for multidimensional pattern extrapolation. In *Advances in Neural Information Processing Systems*, pages 3626–3634, 2014.

[7] Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1775–1784, 2015.

[8] Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423, 2015.

[9] Carl Edward Rasmussen and Christopher KI Williams. Gaussian processes for machine learning. 2006.

[10] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2005.

[11] Andrew Gordon Wilson, Christoph Dann, and Hannes Nickisch. Thoughts on massively scalable gaussian processes. *arXiv preprint arXiv:1511.01870*, 2015.
Algorithm 1 outlines our distillation approach.

**Algorithm 1 Sparse Low-rank Kernel Approximation**

1: **Input:** A well trained kernel function $k_γ$, training feature vectors $X$ and targets $y$, step size $η$, number of iterations $T$ and sparsity $b$.
2: **Output:** Approximated kernel matrix $WK_UU^T$.
3: $U \leftarrow $ K-MEANS$(X)$
4: $K_{XX} \leftarrow k_γ(X,X)$
5: $K_{UU} \leftarrow k_γ(U,U)$
6: $K_{UX} \leftarrow k_γ(X,U)$
7: **Step 1: Initialization**
8: $W \leftarrow 0 \in \mathbb{R}^{n \times m}$
9: for each $x_i$ in $X$ do
10: $J \leftarrow$ indices for $b$ nearest neighbors of $x_i$ in $U$
11: $W_i(J) \leftarrow \arg\min_β ||βK_{UU}(J) - (K_{UX})_i||_2$
12: end for
13: **Step 2: Gradient Descent**
14: for $t = 1$ to $T$ do
15: $E \leftarrow WK_{UU}W^T - K_{XX}$
16: $E_{i,i} \leftarrow 2E_{i,i}$ for $1 \leq i \leq n$
17: $\nabla W \leftarrow E^T WK_{UU}$
18: Project each row of $\nabla W$ to $b$-sparse space
19: Update $W \leftarrow W - η\nabla W$
20: end for

**B Simulation Experiment**

![Figure 2: Kernel Reconstruction Experiments. (a) - (c) Absolute error matrix for reconstructing $K_{XX}$ with kernel distillation, KISS-GP and SoR respectively. (d) $F$-norm error for reconstructing $K_{XX}$ with distillation under different setting of $b$ (sparsity constraint) for $W$.](image)

**Kernel Reconstruction.** We first study how well can kernel distillation reconstruct the full teacher kernel matrix. We generate a $1000 \times 1000$ kernel matrix $K_{XX}$ from RBF kernel evaluated at (sorted) inputs $X$ randomly sampled from $\mathcal{N}(0, 25)$. We compare kernel distillation against KISS-GP and SoR (FITC is essentially SoR with diagonal correction as mentioned in Section 2). We set number of grid points for KISS-GP as 400 and number of inducing points for SoR is set to 200 and kernel distillation to 100. We set the sparsity $b$ to 6 for kernel distillation.

The $F$-norm for errors for are $1.22 \times 10^{-5}$, $8.17 \times 10^{-6}$, $2.39 \times 10^{-7}$ for KISS-GP, SoR and kernel distillation respectively. Kernel distillation achieves lowest $F$-norm error compared to FITC and KISS-GP even the number of inducing points is much fewer for kernel distillation. Moreover, from the absolute error matrices (Figure 2a-c), we can see errors are more evenly distributed for kernel distillation, while there seems to exist a strong error pattern for the other two.

We also show how the sparsity parameter $b$ affect the approximation quality. We evaluate the error with different choices for $b$ as shown in Figure 2(d). We observe that the error converges when the
sparsity $b$ is above 5 in this example. This shows our structured student kernel can approximate the full teacher kernel reasonably well even when $W$ is extremely sparse.

**Toy 1D Example.** To evaluate our distilled model’s predictive ability, we set up the following experiment. We sample $n = 1000$ data points $X$ uniformly from $[-10, 10]$. We set our response $y(x) = \sin(x) \exp(-\frac{x^2}{2}) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 1)$. We train an exact GP with RBF kernel as teacher first then apply kernel distillation with number of inducing points set to 100 and sparsity set to 10. We compare mean and variance prediction of kernel distillation with KISS-GP trained with 400 grid inducing points.

The results are showed in Figure 3. As we can see, mean predictions of kernel distillation are indistinguishable from exact GP and KISS-GP. As for variance, kernel distillation’s predictions are much closer to the variance outputs from exact GP, while the variance outputs predicted by KISS-GP are far away from the exact solution.

This experiment shows a potential problem in KISS-GP, where it sacrifices its ability to provide uncertainty measurements, which is a crucial property of Bayesian modeling, for exchanging massive scalability. On the other hand, kernel distillation can honestly provide uncertainty prediction close to the exact GP model.

**C Experiment Setup**

We compare kernel distillation with teacher kernel (exact GP), FITC as well as KISS-GP. We use the same inducing points selected by K-Means for both FITC and kernel distillation. For KISS-GP, as all the datasets do not lie in lower dimension, we project the input to 2D and construct 2D grid data as the inducing points. Number of inducing points (on 2D grid) for KISS-GP are set to 4,900 (70 per grid dimension) for Boston Housing, 10K for Abalone, 90K for PUMADYM32N, 250K for KIN40K. The number of inducing points for FITC and kernel distillation are 70 for Boston Housing, 200 for Abalone, 1k for PUMADYM32N and KINK40K. The sparsity $b$ in kernel distillation is set to 20 for Boston Housing and 30 for other datasets. For all methods, we choose ARD kernel as the kernel function, which is defined as:

$$k_{\text{ARD}}(x, z) = \exp(-0.5 \sum_{i=1}^{d} (x_i - z_i)^2 / \sigma_i^2)$$

where $d$ is the dimension of the input data and $\sigma_i$’s are the hyper-parameters to learn.

All the experiments were conducted on a PC laptop with Intel Core(TM) i7-6700HQ CPU @ 2.6GHZ and 16.0 GB RAM.