Scalable Gaussian Process for Large Datasets

Hoang Van Do, Emmanuel Vazquez, Tran Quoc Long

Scalable Gaussian Process for Large Datasets

Hoang Van Do
dodobk87@gmail.com
DAC, Viettel Telecom
Hanoi, Vietnam

Emmanuel Vazquez
emmanuel.vazquez@centralesupelec.fr
L2S, CNRS, CentraleSupélec,
Université Paris–Saclay
Gif-sur-Yvette, France

Tran Quoc Long
tqlong@vnu.edu.vn
IAI, VNU University of Engineering
and Technology
Hanoi, Vietnam

ABSTRACT

The domain of Gaussian Processes (GPs), a powerful Bayesian approach in machine learning, emphasizes its scalability for handling large datasets. Traditional GPs, while offering flexibility and a probabilistic interpretation, face computational challenges as dataset sizes increase. This research primarily focuses on a novel method proposed by Noack et al., which introduces "Exact Gaussian Processes for Massive Datasets via Non-Stationary Sparsity-Discovering Kernels"[4], addressing the scalability issues inherent in standard GPs.

KEYWORDS

Gaussian Processes, Scalability, Large Datasets, Non-Stationary, Sparsity-Discovering Kernels, Massive Datasets.

1 INTRODUCTION

A Gaussian Process (GP) is a powerful and flexible probabilistic model used in machine learning for regression and classification tasks. It is particularly useful in scenarios where predictions about the underlying function of your data are desired. In the context of machine learning, GPs are employed for both regression (predicting continuous outputs) and classification (predicting discrete labels). GPs are notably recognized for their ability to provide uncertainty measurements on predictions, a valuable feature in many applications. They find widespread application in fields such as geostatistics, time series analysis, bioinformatics, pattern recognition, machine learning, and others where making predictions with uncertainty estimates is crucial.

Scalable Gaussian Processes are an extension of traditional Gaussian Process models, specifically designed to address the challenges posed by large datasets. While Gaussian Processes offer a powerful and flexible approach to regression and classification, their standard implementation encounters significant computational hurdles as the dataset size increases. In a standard GP, the computational complexity for training is $O(n^3)$, and for prediction, it is $O(n^2)$, where n represents the number of data points. This cubic complexity arises from the necessity to invert the covariance matrix, rendering GPs impractical for large datasets

Central to novel approach is the development of a new class of kernels ultra-flexible, compactly-supported, and non-stationary[4]. These innovative kernels are meticulously engineered to learn and encode a broad spectrum of covariances, including both non-zero and zero covariances. This ability is a marked departure from traditional kernels and is instrumental in uncovering the latent sparse structures within the data. The flexibility of these kernels is key to adapting to the varied and often complex sparse patterns inherent in large datasets, thereby enhancing the effectiveness and accuracy of the Gaussian Process model.

Implementing this novel approach necessitates the use of advanced computational strategies. High-Performance Computing (HPC) is identified as a crucial component to manage the computationally intensive tasks inherent in processing extensive datasets. Additionally, the role of constrained optimization is underscored as a vital mechanism in this framework. Constrained optimization will be employed to optimize the kernel learning process, ensuring that it efficiently captures and represents the sparse structures within the data. This integration of HPC and constrained optimization is expected to not only facilitate the handling of large-scale data but also to refine the overall efficiency and effectiveness of the Gaussian Process model in real-world applications.

2 BACKGROUND AND RELATED WORKS

In the context of real-world applications, the use of standard Gaussian Processes (GPs) can be impractical due to the high memory and computational requirements [5], which grow quadratically and cubically, respectively. This significant time complexity arises from the need to compute matrix inversion, this process involves computational operations of the order $O(n^3)$) and storage requirements of $O(n^2)$, with n representing the count of training points. Given these scaling factors, the model becomes impractical for handling large datasets, as the extensive computations needed for matrix inversion become excessively demanding.

However, this does not completely rule out the application of GPs in big-data domains. To overcome this challenge, various methods have been proposed and developed.

The main goal of these techniques is to bypass the cubic time complexity associated with standard calculations and instead execute them with reduced computational demands, generally in the order of $O(nm^2)$ for time and $O(nm)$ for memory, where m is significantly smaller than n. This approach aims not just to lessen the computational burden, but also to preserve the predictive performance of the standard Gaussian Process model.

The models that address these limitations and facilitate the practical application of standard GP in large-scale scenarios are commonly referred to as scalable GPs. These scalable GPs employ various techniques to manage the computational and memory constraints while striving to deliver accurate and efficient predictions.

The literature acknowledges that there are two primary approaches for managing the substantial computational complexity associated with Gaussian Process models. These approaches can be distinguished based on their focus when it comes to tackling the complexity challenge[2]:

Global Approach: This strategy aims to mitigate the computational complexity on a global scale[3]. It involves techniques that apply to the entire model or dataset. For instance, methods under this category may include sparse approximations that use a subset

of the data to represent the entire dataset, or they might involve mathematical techniques to simplify the overall computational processes. This approach tries to balance the trade-off between computational efficiency and the fidelity of the GP model.

Local Approach: In contrast, the local approach [1] focuses on reducing computational demands at a more granular level. This could involve dividing the dataset into smaller, manageable chunks and applying the GP model to each of these segments individually. Techniques like local kernel approximations or partitioning the input space fall under this category. The local approach often aims to maintain model accuracy while reducing computational burden for each segment of the data.

The choice between these approaches, or a combination of both, depends on the specific requirements and constraints of the application. Some situations may benefit more from a global approach, especially where a broad overview is sufficient, while others might require the precision that a local approach offers.

The fact that some methods combine both global and local strategies highlights the flexibility and adaptability of GP models to various computational and data challenges. This hybrid approach aims to leverage the strengths of both strategies to create a more efficient and effective GP model.

The classification of these approximation types and strategies, as elaborated upon in the cited literature, offers a thorough insight into the means by which scalable Gaussian Process models can be attained. This underscores the continuous dedication within the field towards enhancing the feasibility of GP models for extensive and intricate data analysis endeavors.

3 EXACT GAUSSIAN PROCESSES FOR MASSIVE DATASETS

pproach focuses on defining kernels that are ultra- flexible and capable of learning and representing both non-zero and zero covariances. This concept of exact yet sparse GP, supported by High-Performance Computing (HPC) and constrained optimization, enables scaling exact GP to datasets with over 5 million data points. This paradigm shift offers a more efficient and accurate method for handling large-scale data in Gaussian Process modeling

3.1 Ultra-flexible, compactly-supported, and non-stationary kernel functions

To unearth the inherent sparsity within data, it's crucial to develop a kernel function, denoted as $k(x_1, x_2)$, capable of encoding correlations between data points, including scenarios where such correlations are absent. This kernel must fulfill three key criteria[4]:

1. Compact Support: The kernel should be compactly supported to effectively identify zero covariances. This attribute is essential because our goal is to detect instances where covariance is nonexistent.

2. Non-Stationarity: While compactly-supported kernels have been utilized previously, they have predominantly been in stationary contexts. However, in stationary scenarios, sparsity is exploited only locally, meaning zero covariance is recognized only when a point is significantly distant from others. These kernels fail to learn more intricate, distance-independent patterns of sparsity.

3. Flexibility: The kernel needs the capability to discern varying spatial relationships, recognizing when neighboring points might be correlated or when distant points are uncorrelated, and vice versa. The development of kernels that combine compact support, non-stationarity, and flexibility is crucial for effectively learning existing and non-existing covariances in data. Let's examine the given examples to understand these concepts better.

3.2 High-performance computing to take advantage of sparse Kernel

In this novel approach for scalable Gaussian Processes (GPs), high-performance computing plays a vital role, particularly in handling flexible, non-stationary, and compactly-supported kernels. Initially, the covariance matrix is computed in a dense format to fully utilize multi-threading capabilities. However, for large datasets, this approach could exceed available RAM, making it impractical. Conversely, directly computing the covariance matrix in a sparse format would lead to significant inefficiencies. To circumvent these limitations, we adopt a strategy of initially defining a "host" covariance matrix on a single machine as sparse. We then distribute the computation of dense sub-matrices across multiple nodes in a network, subsequently converting these dense sub-matrices into a sparse format. These sparse sub-matrices are then com- municated back to the host machine and integrated into the original host covariance matrix. This method effectively addresses memory constraints by distributing the memory load of the covariance matrix across numerous computational resources. Furthermore, it allows for the utilization of out-of-core techniques, such as disk storage, when necessary. In terms of computational speed, this approach benefits from the combined power of heterogeneous computing architectures, including GPUs, which are highly efficient in data-parallel operations, and CPUs, adept at threading and task-parallel operations. By distributing memory load and harnessing parallel processing capabilities across different hardware components, our algorithm gains the ability to handle extremely large datasets. The scale of datasets that can be managed is practically boundless, contingent on the availability of sufficient distributed computing resources and the inherent sparsity within the data. This distributed computing strategy not only mitigates the memory limitations but also significantly accelerates the computation process. It allows for the effective handling of the large-scale data typically encountered in extreme-scale GP applications

3.3 Augmented and constrained optimization

The third component of our framework involves an enhanced optimization process, integrating both constrained and augmented strategies. We utilize the kernel design Ultra-flexible, compactly-supported, and non-stationary kernel functions , applying a constrained approach to manage the sparsity level s below a predetermined threshold. This constraint ensures the Gaussian Process (GP) remains precise until the limit of available RAM is reached, at which point the GP transitions to an approximate form autonomously, without requiring user intervention for data point selection.

4 EXPERIMENTS

Our experiments utilized four datasets: a 1-Dimensional generative dataset with a bump function, and three multidimensional publicly available datasets, namely The Weather Stations (reflecting US climate data), the 3D Road Network, and the Bike Sharing dataset. Each of these public datasets features a single target variable and comprises thousands of samples. We used these datasets in a manner that ensures comparability with other studies that have also employed them

The first dataset we experimented is Self-generated one dimension dataset (1D dataset) that consists is 21,000 points were generated from evenly spaced values in the range [0-1], with the y represented by the bump function $f(x) = sin(5x) + cos(10x) + 2(x-0.4)^2 cos(100x)$ adding with some noisy data. $y = f(x) + noise$

The 3D Road Network dataset, obtained from the UCI repository [https://archive.ics.uci.edu/dataset/246/3d+road+network+north+jutland+denmark], is well-known and frequently used in the machine learning community, is a widely recognized collection in the machine learning field. This dataset encompasses 434,874 highly accurate height measurements above sea level, specifically in the North Jutland area of Denmark. It's noted for its relevance in applications like eco-routing and designing bicycle paths. Additionally, it's well-suited for regression analyses. In our usage, longitude and latitude serve as the independent variables, while altitude is the dependent variable.

The Bike sharing dataset, also sourced from the UCI repository [https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset] which contains the hourly count of rental bikes between years 2011 and 2012 in Capital bikeshare system with the corresponding weather and seasonal information. The dataset consists of 17,379 rows and 17 fields, among which: 13 fields are features, 1 field is an ID, 3 fields are labels

Last dataset we experimented is the weather stations across the continental United States [4]. The dataset includes station data and temperature data measured at each station. The dataset contains about 51 million records with values for latitude, longitude, and temperature (in degrees Celsius). For temperature distribution by station, a random sample of 4000 records is taken from the 51 million to represent on a chart according to latitude, longitude, and temperature (in degrees Celsius), as shown in the image below:

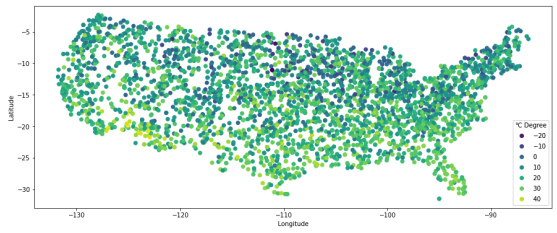

Figure 1: The dataset consists of recordings of the highest daily temperatures, expressed in degrees Celsius (∘C), observed at various locations throughout the United States (N = 4000)

Evaluation Metrics

In assessing the performance of the regression model throughout our experiments, we utilized metrics Root Mean Squared Error (RMSE) and training time. RMSE served as the primary metric for this evaluation, a standard in assessing regression models' predictive accuracy. The RMSE metric is given by the formula:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{1}$$

where $\hat{y}_i$ represents the predicted value, y_i denotes the actual value, and n is the count of predicted samples. In essence, the RMSE is the mean of the squared differences between the predicted and actual values. This part outlines the methods employed to achieve the results, which are summarized in figures and tables, accompanied by visual representations and potential interpretations. The top-performing models underwent ten rounds of training and were assessed on test data. The primary goal of these experiments was to evaluate the accuracy of each model's predictions and to identify GPs that exhibited exceptional performance.

The best models' performances and training durations for each dataset are detailed in some figures below. Every columns in these figures reflects the average value of the corresponding metric, derived from aggregating results from multiple training and evaluation cycles of the top model. To ensure clarity and consistency in presentation, all tables and visualizations showcasing the models' performance adhere to specific formatting guidelines. For formatting purposes, in every visualisation demonstrating the models' performance, the Exploiting natural sparsity and advanced kernel design method of Noack et al. will be referred to as fvGP which will compare with five other methods: SGPR, SVGP, FITC, KISS_GP and BBMM.

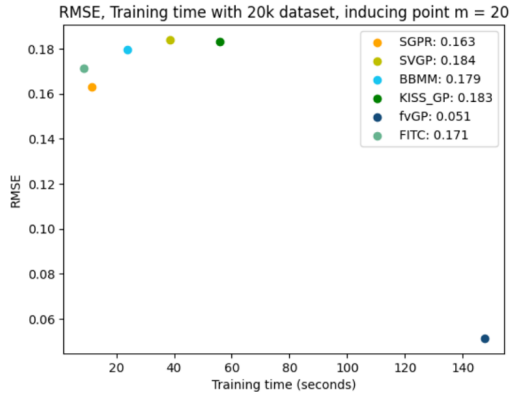

Figure 2: Comparative results of different scalable GPs methods on the seft-generated dataset

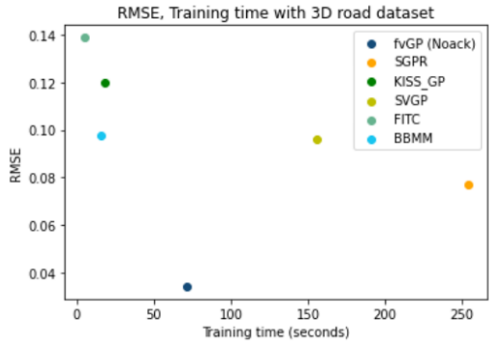

Figure 3: Comparative results of different scalable GPs methods on 3D Road Network dataset

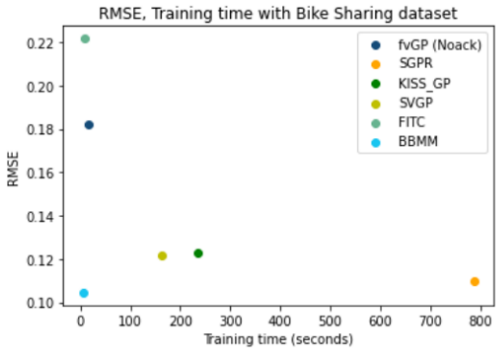

Figure 4: Comparative results of different scalable GPs methods on Bike sharing dataset

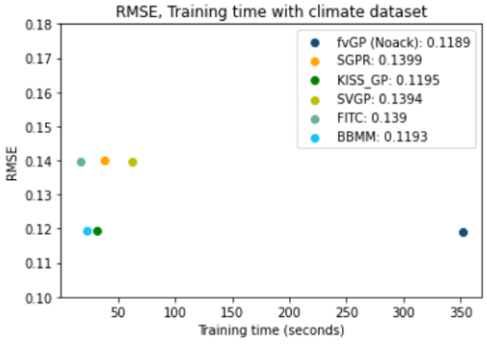

Figure 5: Comparative results of different scalable GPs methods on the US's climate dataset

We observe that 3 out of 4 datasets (1D self-generated, 3D road network, and US climate) exhibit good accuracy, as indicated by the lowest RMSE when using the fvGP method. However, the training time for this method is the highest among the 4 datasets, with 2 out of 4 having the longest duration. This can be attributed to the fvGP method being executed in parallel on two PCs with CPUs. In contrast, other methods were run on GPUs using platforms like Kaggle or Google Colab, which may contribute to their comparatively shorter training times

With the 1-D dataset, a clear trade-off between model accuracy and training time is evident. The fvGP method, despite achieving the lowest RMSE and thus indicating high accuracy, also demands a relatively longer training time compared to other methods.

With the bike-sharing dataset, combining insights from both charts, it is apparent that the fvGP method exhibits good training time. However, its accuracy is only slightly smaller than FITC and is not as favorable when compared to other methods. This discrepancy can be attributed to the use of a random linear algebra (RLA) optimizer, which is not well-suited for small datasets.

With the 3D Road Network dataset, it is evident that the fvGP (Noack) method delivers the highest accuracy and moderate training time. This method seems to strike a favorable balance between accuracy and training time efficiency.

With the Climate dataset, the fvGP method exhibits the lowest RMSE, indicating the most accurate predictions among the compared methods. However, it is noteworthy that the fvGP method takes the longest time to train. Despite being the most accurate, it is not the fastest in terms of training time.

5 CONCLUSION

In summary, this paper represents a significant advancement in the field of Gaussian processes, particularly in the context of massive datasets. We have introduced a comprehensive framework that not only addresses the computational challenges associated with exact GPs but also exploits the natural sparsity present in modern datasets. This approach opens new avenues for applying exact GPs to real-world problems at an unprecedented scale. As we conclude this thesis, we emphasize the practicality and significance of our contributions. Our methodology has the potential to revolutionize the way GPs are employed in various fields, enabling researchers and practitioners to extract valuable insights from massive datasets efficiently. We look forward to further developments and applications of this approach in the future, as it promises to unlock new possibilities in data-driven research and analysis.

REFERENCES

[1] Davit Gogolashvili, Bogdan Kozyrskiy, and Maurizio Filippone. *Locally Smoothed Gaussian Process Regression.* 2022. arXiv: 2210.09998 [stat.ML].

[2] Haitao Liu et al. *When Gaussian Process Meets Big Data: A Review of Scalable GPs.* 2019. arXiv: 1807.01065v2 [stat.ME].

[3] Schweidtmann Artur M. et al. "Deterministic global optimization with Gaussian processes embedded". In: *Mathematical Programming Computation* 13.3 (June 2021), pp. 553–581. ISSN: 1867-2957. DOI: 10.1007/s12532-021-00204-y. URL: http://dx.doi.org/10.1007/s12532-021-00204-y.

[4] Marcus M. Noack et al. "Exact Gaussian processes for massive datasets via non-stationary sparsity-discovering kernels". In: *Scientific Reports* (2023).

[5] Jie Wang. *An Intuitive Tutorial to Gaussian Processes Regression.* 2022. arXiv: 2009.10862 [stat.ML].