International Journal of Mathematics and
Computer Science, **17**(2022), no. 2, 853–864

Gaussian process regression using partition resampling model

Sri Winarni[1], **Sapto Wahyu Indratno**[1,2], **Kurnia Novita Sari**[1]

[1]Statistics Research Group
Faculty of Mathematics and Natural Sciences
Institut Teknologi Bandung
Bandung, Indonesia

[2]University Center of Excellence on Artificial Intelligence
for Vision, Natural Language Processing and Big Data Analytics
Institut Teknologi Bandung
Bandung, Indonesia

email: sriwinarni@s.itb.ac.id, sapto@math.itb.ac.id, kurnia@math.itb.ac.id

(Received November 5, 2021, Accepted December 16, 2021)

Abstract

Gaussian process regression is an interesting probabilistic nonparametric regression method. This method is flexible in describing data patterns. Unfortunately, in the case of big data, this method has expensive computational complexity because it involves inverse matrix operations. In this study, we introduce a partition resampling method to overcome the computational complexity by reducing the dimensions of the data. The purpose of this study is to obtain a Gaussian process regression model using the partition resampling method. This method builds a model from a portion of the data. Although the model uses some data, it has the same advantages as the overall data model. This method partitions the training data and then takes one random sample from each partition to obtain data of a certain size sample. The Gaussian process regression model is a collection of functions that have hyperparameter characteristics. This study predicts new data using a

Key words and phrases: Gaussian processes regression, partition resampling model, computation complexity, hyperparameter, stock price prediction.
Sapto Wahyu Indratno is the corresponding author.
AMS (MOS) Subject Classifications: 62P05, 90C90.
ISSN 1814-0432, 2022, http://ijmcs.future-in-tech.net

hyperparameter averaging approach. We apply our method on stock price data to show that the sample size of 60% with 20 replications produces a close MAPE value of the overall data model. The computation time required is five times faster than the overall data model. So, in this case, the partition resampling method can overcome the computational complexity by reducing the data dimensions to 60%. In other cases, this method can be used as an alternative to model time series data more effectively.

1 Introduction

Time series prediction is an urgent issue because it has a lot to do with decisions. In the financial sector, stock price predictions influence the decision to buy or sell shares. Stock price data often has a fairly complex pattern and so that the linear model approach often does not provide representative results. This study applies Gaussian process regression to predict stock price data. The Gaussian process is a generalization of the Gaussian probability distribution [1]. The Gaussian process parameters are the mean function and the covariance matrix. One of the most commonly used covariance (kernel) functions is the square exponential. The dimension of the parameter function will increase as the amount of data used increases. This Gaussian process can follow the data pattern flexibly. Another advantage of the Gaussian process is that it produces a prediction of the output value and its distribution in the form of a confidence interval. Statistically, the interval estimator is better than the point estimator. This is what distinguishes it from other predictive models. Gaussian process regression is widely used in big data analysis that has complex data patterns.

In this study, we construct a Gaussian process regression model using the quadratic exponential covariance function. The function has hyperparameters; namely, the length scale and signal variance. Optimizing the hyperparameter value using the maximum likelihood parameter estimation method involves calculating the inverse of the covariance matrix with a computational complexity of $O(n^3)$ for data of size n. If the data size is larger, then the computational complexity becomes more expensive and the memory and time requirements will also increase. The most commonly used method to overcome this limitation in computational complexity is the sparse approach.

The starting point to any of the approximations is a set of inducing inputs to build a low approximation to the covariance matrix, [2], [3], [4], [5]. Other approaches are the sparse pseudo-input Gaussian process (SPGP) [6],

local sparse [7] [8], Variational formulation with inducing variables [9], and stochastic variational [10]. The Gaussian process of the sparse approach with the variational learning method using inducing variables can reduce the complexity of calculations from $O(n^3)$ to $O(nm^3)$, where m is the number of inducing variables ($m < n$). Stochastic variational inference reduces complexity computation to $O(m^3)$. The above methods pick a subset of data (inducing variables) based on theoretical considerations. Another approach is an algorithm that combines estimates from models developed using subsets of the data obtained in a manner similar to the bootstrap [11], bagging (bootstrap averaging) method in industrial chemical plant [12], and a biological application [13]. This research is a development of a resampling technique used by [14] that uses move resampling to reduce the complexity of computation.

The motivation of this paper is to overcome the computational complexity by reducing the dimensions of the data. We propose a partition resampling method to build a model using a portion of the complete data. This method partitions as many as m of the overall data size n. Then take one random sample from each partition to get a sample of size m. This procedure is repeated k times. Each sample estimates the optimum hyperparameter and predicts the new data using the average k hyperparameters. It is urgent to conduct this research to obtain an iterative sample partitioning model that has a level of accuracy close to the complete model. It is also important to get a good predictive model using only sample data (not all data). This is useful for reducing the dimensions of the covariance matrix so that it can solve the problem of computational complexity in the Gaussian regression process. The question is what data size can produce a partition model with a good degree of accuracy. The purpose of this paper is to obtain a Gaussian process regression model with a resampling partition approach that gives a good accuracy value.

2 Algorithm of partition resampling Gaussian processes regression

Consider the training data $\mathcal{D} = (X, \mathbf{y})$ consisting of an input matrix $X = [\mathbf{x_1 x_2 x_n}]^T$ representing the amount of training data and $\mathbf{y} = [y_1 y_2 y_n]^T$ representing the associated target vector. Every vector $\mathbf{x}_i \in \Re^D (i = 1, 2, ..., n)$ is related to y_i. We use the following partition

$$m = \delta n; 0 < \delta < 1, \tag{2.1}$$

δ is the proportion of the sample size.

The Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution [1]. A Gaussian process is completely specified by its mean function and covariance function. We will write the Gaussian process as

$$f(\mathbf{fx}) \sim \mathbf{GP}(\mathbf{m}(\ \mathbf{bfx}), \mathbf{k}(\mathbf{x}, \mathbf{x}')) \tag{2.2}$$

We define $m(\mathbf{x})$ as the mean function and $k(\mathbf{x}, \mathbf{x}')$ as the covariance function between pairs of random variables $\mathbf{x}$ and $\mathbf{x}'$. In this study, we use the squared exponential covariance function as

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp(-\frac{1}{2l^2}|\mathbf{x} - \mathbf{x}'|^2) \tag{2.3}$$

with σ_f^2 (signal variance) and l (length scale) are hyperparameters. The optimal hyperparameter is obtained by maximizing the likelihood log function as

$$Log(p(y|X)) = -\frac{1}{2}y^T[K(X,X)+\sigma_n^2 I]^{-1}y - \frac{1}{2}log|K(X,X)+\sigma_n^2 I| - \frac{n}{2}log2(\pi), \tag{2.4}$$

where $K(X, X)$ is the covariance matrix of training data input and σ_n^2 is the noise variance, and I is the identity matrix.

Consider the testing data $\mathcal{D}_* = (X_*, \mathbf{y}_*)$ with X_* as the input value of testing and $\mathbf{y}_*$ as the output value of testing. The prediction of the output value $\mathbf{y}_*$ at the testing input value X_* has a Gaussian distribution with the mean parameter μ_* and the variance σ_n^2 as

$$y_*|X, y, X_* \sim \mathcal{N}(\mu_*, \sigma_*^2) \tag{2.5}$$

The mean and variance parameters are

$$\mu_* = K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} y \tag{2.6}$$

$$\sigma_*^2 = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*) \tag{2.7}$$

with $K(X_*, X)$ being the covariance matrix between the testing data with training data input, while $K(X, X_*)$ is the transpose matrix of $K(X_*, X)$. Next, $K(X_*, X_*)$ is the covariance matrix of testing data input. The predictions using average hyperparameters are

$$\bar{l} = \frac{1}{k}\sum_{i=1}^{k} l_i \tag{2.8}$$

and

$$\bar{\sigma}_f^2 = \frac{1}{k}\sum_{i=1}^{k}\sigma_f^2 i \tag{2.9}$$

The data analysis conducted in this paper was to form a partition resampling model and compare it with the overall data model to obtain the best partition model. If the partition model gives good results, then we can use it as an alternative model. The stages of the analysis are carried out in the following algorithm:

Input:

$\mathcal{D}$ is the training dataset of size n
δ is the proportion of partition
k is the repeated partition sample

Process:

Step 1. For $i = 1 : k$
Make m partitions from $\mathcal{D}$ using Equation 2.1
Select a sample from each partition
Estimate the optimum hyperparameter from m using Equation 2.4
Average of m hyperparameters
end
Step 2. Predict output for point $X_* \in \mathcal{D}_*$ using Equation 2.6 and 2.7.

Output:

μ_*: the output predicted value at X_*

Model goodness is measured from the Mean Absolute Percentage Error (MAPE) value in Equation 2.10. MAPE is one measure of whether a model is good or not, measuring the closeness of the predicted value to the true value.

$$MAPE = \frac{1}{t}\sum_{j=1}^{t}|(\mathbf{y_{*j}} - \mu_{*j})/\mathbf{y_{*j}}| \tag{2.10}$$

where μ_{*j} is the predicted output at point X_{*j} (Equation 2.6) and the value of $\mathbf{y_{*j}}$ is the actual value at the point X_{*j}. The best partition model gives the smallest MAPE value. Evaluation is also done by comparing the computation time required for the partition model and the overall data model.

3 Experimentation and results

The training dataset used in this research is closing stock price data of Costco Wholesale Corporation of size 2200 during the period from December 1, 2011 to August 20, 2020 (finance.yahoo.com). The testing point is the stock price on August 31, 2020 with a closing stock price value of 347.66. In this study, an experiment was carried out by testing the proportion of the sample size of 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, and 90%. The repetitions were done 10, 20, 30, 40, and 50 times. The algorithm of hyperparameter average approach produces an estimate of the average length-scale given in Table 1.

Table 1: The results of the average length scale on the sample size and replications

Sample size (%)	Rep10	Rep20	Rep30	Rep40	Rep50
10	5.2776	5.1564	5.1928	5.1591	5.1614
20	3.9388	3.8615	3.9294	3.8882	3.8691
30	3.7352	3.6872	3.6822	3.6947	3.6989
40	3.5435	3.5335	3.5370	3.5403	3.5415
50	3.3489	3.3258	3.3433	3.3613	3.3523
60	3.1787	3.1804	3.1833	3.2043	3.1784
70	2.9451	2.9451	2.9455	2.9456	2.9459
80	2.8745	2.8256	2.8089	2.8013	2.8175
90	2.6335	2.6334	2.6333	2.6333	2.6333

Different sample sizes give different mean length-scale results. The length-scale determines the length of the 'wiggles' in the function. Length-scale describes how smooth a function is. Small length-scale value means that function values can change quickly, large values characterize functions that change only slowly. The more data used, the smaller the length-scale. The less data used in the partition model, the smoother the pattern is. The results of the average length-scale obtained were relatively the same for all the number of replications. The average result of the next hyperparameter is the mean signal variance shown in Table 2.

The variance determines the average distance of the function away from its mean. The fewer data will produce predictions with greater variance. The number of replication gives the average signal variance prediction results that are relatively the same. The hyperparameter average results are used to predict the output value with the results given in Table 3.

Table 3 shows that the larger the sample size, the more the predicted

Table 2: The results of mean signal variance on sample size and replicates

Sample size (%)	Rep10	Rep20	Rep30	Rep40	Rep50
10	5.0503	5.0177	4.9940	4.9853	4.9867
20	4.8018	4.7909	4.7968	4.7937	4.7905
30	4.7844	4.7841	4.7832	4.7837	4.7830
40	4.7757	4.7756	4.7763	4.7761	4.7762
50	4.7607	4.7605	4.7608	4.7617	4.7612
60	4.7506	4.7507	4.7511	4.7546	4.6501
70	4.7420	4.7421	4.7420	4.7420	4.7420
80	4.7385	4.7377	4.7374	4.7373	4.7375
90	4.7324	4.7324	4.7324	4.7324	4.7324

Table 3: The results of mean prediction using hyperparameters average

Sample size (%)	Rep10	Rep20	Rep30	Rep40	Rep50
10	342.0376	343.3752	342.8451	342.9831	343.0439
20	344.7373	342.4115	344.8452	343.8405	342.8757
30	340.9731	342.3189	342.5084	342.0604	341.9833
40	345.2178	345.2839	345.4933	345.3548	345.4332
50	347.4218	347.5967	347.2201	346.9816	347.1247
60	347.6081	347.6600	347.7007	347.6697	347.7003
70	347.6897	347.6897	347.6897	347.6897	347.6897
80	347.8715	347.8715	348.2885	348.4760	348.0735
90	348.7984	348.7972	348.7953	348.7959	348.7960

output results will be closer to the predicted value of the full model. At a sample size of 60%, the predicted output values approximate their actual values. The results of mean prediction obtained were relatively the same for all the number of replications (Table 4.).

If a small amount of data is used, then the resulting variance will be large. The further away the predictor is, the greater the variance. The number of repetitions gives relatively the same variance. The MAPE results for the partition model with the hyperparameters average approach are given in Table 5.

MAPE is the mean or the average of the absolute percentage errors of forecasts. Error is defined as the actual or observed value minus the forecasted value. MAPE is a percentage measure of how accurate a forecast system is. In Table 5, we can see that the MAPE value obtained is relatively small. In the partition model, the MAPE value gets smaller as the sample

Table 4: The results of variance prediction using hyperparameters averages

Sample size (%)	Rep10	Rep20	Rep30	Rep40	Rep50
10	67.5625	63.1615	61.2539	60.0204	59.5526
20	31.4105	30.3203	31.1974	29.9631	30.1750
30	28.1233	27.6996	27.5951	27.9910	28.0388
40	23.7921	23.5681	23.6204	23.6693	23.5880
50	18.4803	18.3376	18.3662	18.4397	18.2700
60	18.4581	18.4857	18.4915	18.5090	18.1413
70	17,5432	17.7865	17.3421	17.8437	17.4378
80	15.1685	15.1685	15.0983	15.0755	15.1320
90	13.5986	13.2764	13.2175	13.2980	13.5648

Table 5: The results of MAPE output prediction using hyperparameters average

Sample size (%)	Rep10	Rep20	Rep30	Rep40	Rep50
10	0.016172	0.012325	0.013849	0.013453	0.013278
20	0.008407	0.015097	0.008096	0.010986	0.013761
30	0.019234	0.015363	0.014818	0.016107	0.016328
40	0.007025	0.006835	0.006232	0.006631	0.006405
50	0.000785	0.000182	0.001265	0.001951	0.001540
60	0.000149	0.000000	0.000117	0.000028	0.000116
70	0.000185	0.000085	0.000085	0.000085	0.000085
80	0.000608	0.000608	0.001808	0.002347	0.001189
90	0.003271	0.003271	0.003266	0.003268	0.003268

size used increases. At a sample size of 60%, the MAPE value is close to zero meaning that the partition resampling approach can produce a good model. In this case, 20 replicates can be performed with small MAPE results.

Several replicates yielded relatively similar MAPE. Evaluation is also carried out with the required computation time. The computation time required to build the full model and make predictions using Intel CORE i5 7th Gen is 166.513 seconds. The results of computational time for partitional resampling model are shown in Table 6.

The computation time required by the partition resampling model is faster than the full model. Computation time increases with increasing sample size and replication. In the partition model with a sample size of 60% and 70%, the obtained MAPE is relatively the same but the computation time is somewhat different. Based on MAPE results and computation time,

Table 6: The results of computation time (seconds) using hyperparameters average

Sample size (%)	Rep10	Rep20	Rep30	Rep40	Rep50
10	5.892	7.765	9.439	10.321	12.136
20	8.167	10.986	12.328	15.679	18.109
30	11.041	12.921	15.126	18.374	23.768
40	14.478	15.834	18.769	23.438	26.908
50	19.804	20.925	25.431	28.189	32.091
60	27.656	28.124	32.876	35.639	37.176
70	57.418	59.153	62.398	65.904	69.980
80	91.132	95.432	98.812	101.765	105.438
90	116.811	119.397	121.484	124.850	134.750

choosing a sample size 60% with 20 replications the output prediction results are given in Fig. 1.

At the time of 2201; namely, on August 31, 2020, the prediction of stock price results were obtained in the form of a normal distribution with an average value of 347.6600 and a variance of 18.4857. The full Gaussian process model uses all of the data to create the model. Using Equation 3 as a covariance function and optimization using Equation 4, the optimum hyperparameter l is 2.5979 and σ_f^2 is 4.7263. This value is used to predict the average and variance output value using Equations 2.6 and 2.7. The prediction results are given in Fig. 1.

At the 2201 stage; namely, on August 31, 2020, the prediction of stock price results was obtained in the form of a normal distribution with an average value of 348.8567 and a variance of 12.0618. The confidence interval formed with 2 standard deviations is the lower limit of 341.9106 and the upper limit of 355.8027

The prediction results using the partition model with a sample size of 60% and 20 replication gave a smoother data pattern than the full model. The hyperparameter average value obtained was greater. The predicted value at the 2201 time series resulted in a smaller MAPE but a greater variance value. This model can be used as an alternative with the computation time required being also smaller than the overall data model.

The results of this initial research are numerical. This study provides information that the resampling partition method can be an alternative in forming an effective Gaussian process regression model. Our next research is to obtain theoretical results related to the size of the sample partition that

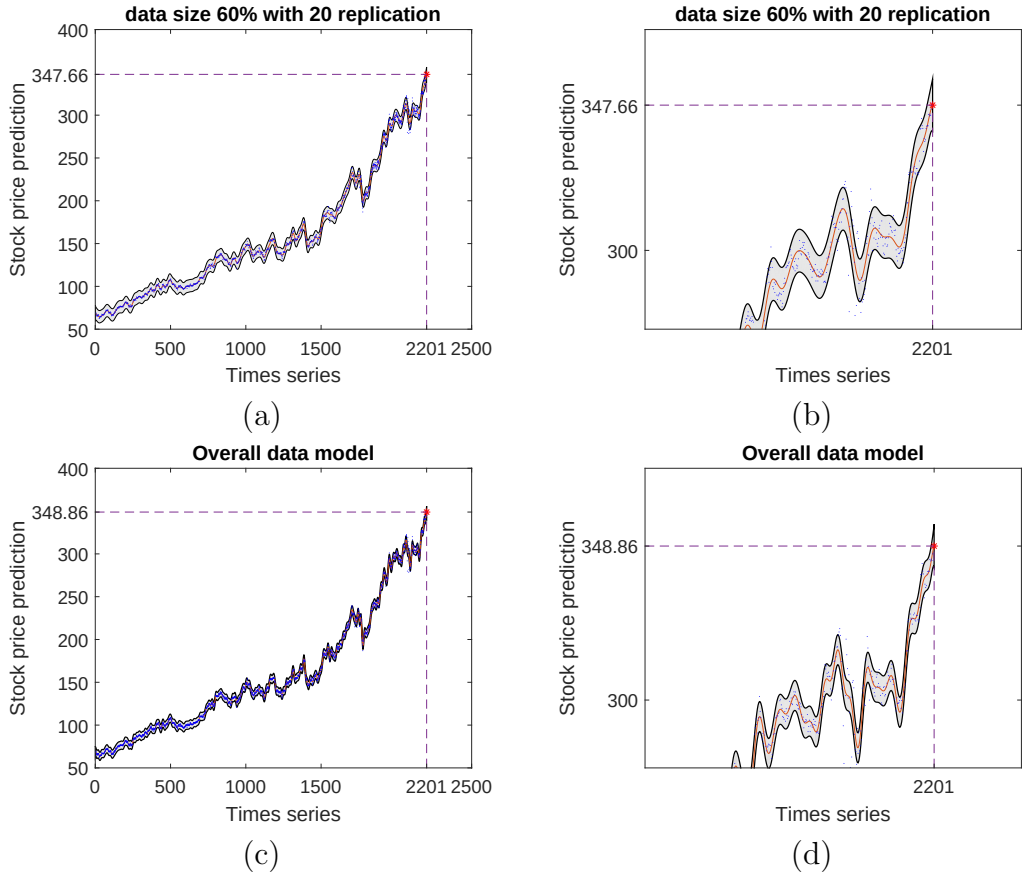

Figure 1: The output prediction results in partition resampling model using a sample size of 60% with 20 replications (a), and the predicted output magnification for the partition model (b). The output prediction results in the full model (c) and the predicted output magnification for the full model (d). Point symbols are actual data and the inner line is the prediction result of the training data. The gray area is the confidence interval for the predicted output.

can provide good predictive results so that it can be applied to other data cases.

4 Conclusion

In the case of Costco Wholesale Corporation stock price data, a partition resampling model using the hyperparameter averaging approach can reduce the data dimensions so that it can overcome the computational complexity of the full model. This reduction in computational complexity will of course also reduce the costs and time required for modeling. The partition model with a sample size of 60% and 20 replication can be used as an alternative in modeling and predicting stock prices. The predicted results obtained are close to the actual value with a variance value that is not much different from the full model. In addition, the computation time required is also faster. This 60% data size does not necessarily apply to other data. This study provides information that the partition resampling method can be an effective alternative model.

Acknowledgment. The authors thank the ITBs Program of Research 2021, Faculty of Mathematics and Natural Sciences, entitled "Risk Statistics-Based Information System in an Effort to Support Bandung City Government Policies Related to Pandemic Handling" for supporting funds.

References

[1] C. E. Rasmussen, C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[2] M. Seeger, C. K. I. Williams, N. D. Lawrence, "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression", *Proc. Int. Conf of workshop on AI and Statistics*, (2003).

[3] J. Q. N. Candela, C. E. Rasmussen, "Unifying view of sparse approximate Gaussian process regression", *J. of Machine Learning Research*, **6,** (2005).

[4] J. Schereiter, D. N. Tuong, M. Toussaint, "Efficient sparsification for Gaussian process regression", *J. Neurocomputing,* **192**, (2016), 29–37.

[5] E. L. Snelson, ,*Flexible and efficient Gaussian process models for machine learning*, dissertation University of London (2007).

[6] E. L. Snelson, Z. Ghahramani, "Sparse Gaussian Processes using Pseudo-inputs", *J. Neural Information Processing Systems*, (2006).

[7] E. L. Snelson, Z. Ghahramani, "Local and global sparse Gaussian process approximations", *Proc. Int. Conf. on Artificial Intelligence and Statistics PMLR,* **2**, (2007), 524–531.

[8] D. N. Toung, J. Peter, M. Seeger, "Local Gaussian Process Regression for Real Time Online Model Learning and Control", *Proc. Int. Conf. Neural Information Processing Systems*, (2008), 1193–1200.

[9] M. K. Titsias, J. Peter, M. Seeger, "Variational Learning of Inducing Variables in Sparse Gaussian Processes", *Proc. Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, (2009).

[10] J. Hensman, N. Fusi, N. D. Lawrence, "Gaussian processes for big data", *Proc. Int. Conf.uncertainty in Artificial Intelligence*, (2013), 282–290.

[11] S. Das, S. Roy, R. Sambasivan, "Fast Gaussian process regression for big data", *J. Big Data Research,* **14**, (2018), 12–26.

[12] T. Chen, J. Ren, "Bagging for Gaussian process regression", *J. Neurocomputing*, (2009), 1605–1610.

[13] P. D. W. Kirk, M. P. H. Stumpf, "Gaussian process regression bootstrapping: exploring the effects of uncertainty in time course data.", *J. Bioinformatics,* **25,** no. 10, (2009).

[14] S. Winarni, S. W. and Indratno, "Gaussian Process Regression with Average Hyperparameter", *J. Phys. : Conf. Series*, (2020), 1–10.