The pitfalls of using Gaussian Process Regression for normative modeling

Bohan Xu[1,2,*], Rayus Kuplicki[1], Sandip Sen[2], Martin P. Paulus[1,3,4]

1 Laureate Institute for Brain Research, Tulsa, OK, United States
2 Department of Computer Science, Tandy School of Computer Science, University of Tulsa, Tulsa, OK, United States
3 Department of Community Medicine, Oxley College of Health Sciences, University of Tulsa, Tulsa, OK, United States
4 Department of Psychiatry, School of Medicine, University of California San Diego, San Diego, CA, United States

* E-mail: bxu@laureateinstitute.org

Abstract

Normative modeling, a group of methods used to quantify an individual's deviation from some expected trajectory relative to observed variability around that trajectory, has been used to characterize subject heterogeneity. Gaussian Processes Regression includes an estimate of variable uncertainty across the input domain, which at face value makes it an attractive method to normalize the cohort heterogeneity where the deviation between predicted value and true observation is divided by the derived uncertainty directly from Gaussian Processes Regression. However, we show that the uncertainty directly from Gaussian Processes Regression is irrelevant to the cohort heterogeneity in general.

Introduction

In case-control studies, participants are assigned labels and classified into one or more categories based on their similarities or common criteria, with little consideration for the heterogeneity within each cohort. Meanwhile, normative modeling is becoming increasingly popular. In a normative model, each observation is quantified as a normalized deviation with respect to the cohort heterogeneity. The growth chart [1, 2] is an example normative model as shown in Fig. 1, where a series of percentile curves (normalized deviation) illustrate the distribution of selected body measurements in children. Another widely-used measure for normalized deviation is the z-score, which is calculated by dividing the difference between an observation and the reference model, i.e., residual, by a standard deviation that represents local heterogeneity and assumes residuals are Gaussian distributed locally.

Fig 1. Weight-for-Age Boys: Birth to 2 years [3]. The percentiles show the distribution of weights in boys form birth to 2 years. Black dots: observations; red error bars: epistemic uncertainty; blue curly brackets: aleatoric uncertainty.

The uncertainty sometimes can be classified into two categories: epistemic and aleatoric uncertainties. Epistemic uncertainty is known as systematic uncertainty and is due to things one could in principle know but do not in practice; aleatoric uncertainty is known as statistical uncertainty and is representative of unknowns that differ each time we run the same experiment [4]. Epistemic uncertainty is often introduced by the limited dataset size and can be reduced by adding more observations. On the other hand, aleatoric uncertainty represents a character of heterogeneity in the underlying distribution itself which is unrelated to sample size, so it cannot be reduced by modifying the dataset, and this is the heterogeneity a normative model should measure. As shown in Fig. 1, larger number and density of data points (black dots) reduce the epistemic uncertainty (red error bars), while the aleatoric uncertainty (blue curly brackets) is unrelated to the sample size or distribution. The confidence intervals obtained from most statistical tests and advanced machine learning models only capture epistemic uncertainty, while a normative model is designed to capture the aleatoric uncertainty.

Gaussian Process Regression (GPR) has been widely used in many domains. Schulz et al. [5] presented a tutorial on the GPR with the mathematics behind the model as well as several applications to real-life datasets/problems. Tonner et al. [6] developed a GPR based model and testing framework to capture the microbial population growth and shown their proposed approach outperformed primary growth models. Banerjee et al. [7] and Raissi et al. [8] introduced two novel approaches to improve the efficiency of GPR in “big data” problems.

However, some previous research implemented the GPR as a normative modeling approach and utilized the derived prediction variance to model the cohort heterogeneity. Ziegler et al. [9] attempted to build a normative model for diagnosing mild cognitive impairment and Alzheimer’s disease based on the normalized deviation of predicted brain volume from GPR. Marquand et al. [10] used delay discounting as covariates and reward-related brain activity derived from task Functional Magnetic Resonance Imaging (fMRI) as the target variable with GPR and extreme value statistics to identify the participants with Attention-Deficit/Hyperactivity Disorder (ADHD). Wolfers et al. [11] investigated the deviation of brain volume in an ADHD cohort from healthy control group (HC) with respect to age and gender, and they also explored the heterogeneous phenotype of brain volume for schizophrenia and bipolar disorder with GPR [12]. Zabihi et al. [13] studied Autism Spectrum Disorder (ASD) regarding the deviation of cortical thickness via a similar methodology.

In this paper, we introduce some background knowledge related to GPR. We then present a rigorous mathematical derivation and several examples to demonstrate that the variance from GPR cannot be used in a normative model alone. In the last section, we discuss the difficulties and disadvantages of modeling the cohort heterogeneity by modifying original GPR variance, and a misunderstanding existed in previous research.

Materials and methods

Gaussian Process Regression

The relation between the observation and the predictive model usually can be expressed as

$$y = f(\boldsymbol{x}) + \varepsilon, \tag{1}$$

where y is the observation (output), $f(\cdot)$ represents the predictive model, $\boldsymbol{x}$ is a vector of independent variables (input) corresponding to the output y, and ε is the noise term which follows a normal distribution $\varepsilon \sim \mathcal{N}\left(0,\sigma^2_{\text{noise}}\right)$. Gaussian Process Regression (GPR) assumes a zero-mean normal distribution over the predictive model

$$f(\cdot) \sim \mathcal{N}(0,k(\cdot,\cdot)), \tag{2}$$

where $k(\cdot,\cdot)$ is some covariance (kernel) function. Given the training set input $\boldsymbol{X}$ and testing set input $\boldsymbol{X}_*$, since both of them follow the same distribution, we have

$$f\left(\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{X}_*\end{bmatrix}\right)\sim\mathcal{N}\left(\mathbf{0},\begin{bmatrix}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) & \boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}_*)\\ \boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X}) & \boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X}_*)\end{bmatrix}\right). \tag{3}$$

According to the Eq. 1, the observation follows the summation of these two normal distributions

$$\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{y}_*\end{bmatrix}=f\left(\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{X}_*\end{bmatrix}\right)+\begin{bmatrix}\boldsymbol{\varepsilon}\\ \boldsymbol{\varepsilon}_*\end{bmatrix}\sim\mathcal{N}\left(\mathbf{0},\begin{bmatrix}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})+\boldsymbol{\Sigma}^2_{\text{train}} & \boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}_*)\\ \boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X}) & \boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X}_*)+\boldsymbol{\Sigma}^2_{\text{test}}\end{bmatrix}\right), \tag{4}$$

where $\boldsymbol{\Sigma}^2_{\text{train}}$ and $\boldsymbol{\Sigma}^2_{\text{test}}$ are two diagonal matrices that represent the variance of observation noise in training and testing sets, and the diagonal elements equal σ^2_{noise}. By the rules of conditional Gaussian distribution, the prediction of testing set $\boldsymbol{y}_*$ follows a normal distribution $\boldsymbol{y}_*\sim\mathcal{N}(\boldsymbol{\mu}_*,\boldsymbol{\Sigma}^2_*)$, where $\boldsymbol{\mu}_*$ and $\boldsymbol{\Sigma}^2_*$ are defined as [14, 15]

$$\boldsymbol{\mu}_*=\boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X})\left[\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})+\boldsymbol{\Sigma}^2_{\text{train}}\right]^{-1}\boldsymbol{y}, \tag{5a}$$

$$\boldsymbol{\Sigma}^2_*=\boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X}_*)+\boldsymbol{\Sigma}^2_{\text{test}}-\boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X})\left[\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})+\boldsymbol{\Sigma}^2_{\text{train}}\right]^{-1}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}_*). \tag{5b}$$

Kernel Trick

Similar to Support Vector Machines (SVM), the kernel trick can also be implemented with GPR to project the input of data from the original space into a same or higher dimensional feature space via some mapping function $z(\cdot)$. Given a pair of inputs $(\boldsymbol{x}_1,\boldsymbol{x}_2)$, the kernel function calculates the inner product of the coordinates in the feature space, i.e., $k(\boldsymbol{x}_1,\boldsymbol{x}_2)=\boldsymbol{z}(\boldsymbol{x}_1)\boldsymbol{z}(\boldsymbol{x}_2)^T$ [16, 17]. The kernel trick avoids the expensive computation of calculating the coordinate in the feature space for each input. We use the linear kernel and Radial Basis Function kernel (RBF) as examples to illustrate this advantage.

Linear Kernel

The linear kernel is non-stationary and the simplest kernel, which is defined as

$$k(\boldsymbol{x}_1,\boldsymbol{x}_2)=\boldsymbol{x}_1\boldsymbol{x}_2^T, \tag{6a}$$

$$z(\boldsymbol{x})=\boldsymbol{x}, \tag{6b}$$

where the input is projected into a feature space according to Eq. 6b, and the feature space is the original space.

Radial Basis Function Kernel

The RBF kernel is a stationary kernel, which is also widely used and defined as [17]

$$k(\boldsymbol{x}_1,\boldsymbol{x}_2)=e^{-\frac{\|\boldsymbol{x}_1-\boldsymbol{x}_2\|^2}{2l^2}}, \tag{7a}$$

$$z(\boldsymbol{x})=\left[\frac{e^{-\frac{\|\boldsymbol{x}\|^2}{2l^2j}}}{\sqrt{l^{2j}j!}^{\frac{1}{j}}}\sqrt{\frac{j!}{n_1!\cdots n_k!}}x_1^{n_1}\cdots x_k^{n_k}\right]_{j=0,\cdots,\infty,\sum_{i=1}^{k}n_i=j}, \tag{7b}$$

where l is a free scaling parameter. The RBF kernel projects the input from the original space onto an infinite dimensional feature space where the mapping is defined by Eq. 7b. It is impossible to exactly compute the coordinates in an infinite dimensional space, while Eq. 7a still allows straightforward computation of the inner product for coordinate pairs in that feature space.

Estimated Uncertainty for GPR

One benefit of using GPR to build a data-driven model is the predictions are associated with the derived variances as shown in Eq. 5. However, we need to emphasize that this variance is only related to the kernel function $k(\cdot,\cdot)$ and distribution/coordinate of training set input $\boldsymbol{X}$, i.e., it cannot be utilized in a normative model approach alone to capture the variance introduced by the conditional distribution $Var(y|\boldsymbol{x})$.

We better illustrate and verify this statement through simplifying the Eq. 5b. Since any kernel function $k(\cdot,\cdot)$ can be written as the inner product of a coordinate pair in the feature space by some mapping function $z(\cdot)$, we present our derivation in a general format. We define a variable $\boldsymbol{x}_*$ which represents a testing input, then Eq. 5b can be written as[1]

$$\begin{aligned}&\Sigma_*^2(\boldsymbol{x}_*)\\ &=k(\boldsymbol{x}_*,\boldsymbol{x}_*)+\sigma_{\text{test}}^2-\boldsymbol{k}(\boldsymbol{x}_*,\boldsymbol{X})\left[\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})+\boldsymbol{\Sigma}_{\text{train}}^2\right]^{-1}\boldsymbol{k}(\boldsymbol{X},\boldsymbol{x}_*) && \text{(8a)}\\ &=\boldsymbol{z}(\boldsymbol{x}_*)\boldsymbol{z}(\boldsymbol{x}_*)^T+\sigma_{\text{test}}^2-\boldsymbol{z}(\boldsymbol{x}_*)\boldsymbol{Z}(\boldsymbol{X})^T\left[\boldsymbol{Z}(\boldsymbol{X})\boldsymbol{Z}(\boldsymbol{X})^T+\boldsymbol{\Sigma}_{\text{train}}^2\right]^{-1}\boldsymbol{Z}(\boldsymbol{X})\boldsymbol{z}(\boldsymbol{x}_*)^T. && \text{(8b)}\end{aligned}$$

Applying Singular Value Decomposition (SVD) on $\boldsymbol{\Sigma}_{\text{train}}^{-1}\boldsymbol{Z}(\boldsymbol{X})=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, Eq. 8 is reformulated as[2]

$$\begin{aligned}&\Sigma_*^2(\boldsymbol{x}_*)\\ &=\boldsymbol{z}(\boldsymbol{x}_*)\boldsymbol{z}(\boldsymbol{x}_*)^T+\sigma_{\text{test}}^2-\boldsymbol{z}(\boldsymbol{x}_*)\boldsymbol{Z}(\boldsymbol{X})^T\left[\boldsymbol{Z}(\boldsymbol{X})\boldsymbol{Z}(\boldsymbol{X})^T+\boldsymbol{\Sigma}_{\text{train}}^2\right]^{-1}\boldsymbol{Z}(\boldsymbol{X})\boldsymbol{z}(\boldsymbol{x}_*)^T\\ &=\underbrace{\boldsymbol{z}(\boldsymbol{x}_*)\boldsymbol{V}\left[\boldsymbol{I}-\boldsymbol{\Sigma}^T\left(\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T+\boldsymbol{I}\right)^{-1}\boldsymbol{\Sigma}\right]\boldsymbol{V}^T\boldsymbol{z}(\boldsymbol{x}_*)^T}_{\text{quadratic term}}+\underbrace{\sigma_{\text{test}}^2}_{\text{constant}}.\end{aligned}\tag{9}$$

After simplification, the variance is reformulated as Eq. 9, which is a summation of a quadratic term for $\boldsymbol{z}(\boldsymbol{x}_*)$ and a constant represents the noise, $\boldsymbol{\Sigma}$ and $\boldsymbol{V}$ are constant matrices where the values are fully depended on training input $\boldsymbol{X}$, training noise $\boldsymbol{\Sigma}_{\text{train}}$, and mapping function $z(\cdot)$ or kernel function $k(\cdot,\cdot)$.[3]

Modification of Uncertainty from GPR

Regarding Eq. 9, the variance calculated via Eq. 5b is purely depended on kernel function and training data input, thus it is only able to capture the epistemic uncertainty which could be reduced by modifying or adding training data. The derived variance from GPR could be extended to model the heterogeneity $Var(y|\boldsymbol{x})$ for a normative model by adding an aleatoric variance term into Eq. 5b

$$\begin{aligned}Var(\boldsymbol{y}_*|\boldsymbol{X}_*)=&\underbrace{\boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X}_*)+\boldsymbol{\Sigma}_{\text{test}}^2-\boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X})\left[\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})+\boldsymbol{\Sigma}_{\text{train}}^2\right]^{-1}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}_*)}_{\text{epistemic uncertainty}}\\ &+\underbrace{\boldsymbol{\Sigma}_{\text{aleatoric}}^2(\boldsymbol{X}_*)}_{\text{aleatoric uncertainty}},\end{aligned}\tag{10}$$

where $\boldsymbol{\Sigma}_{\text{aleatoric}}^2(\boldsymbol{X}_*)$ represents the data character of heterogeneity in output at given locations on the input space. This formula, however, is not implemented in any previous research as we know and we will discuss the difficulties and disadvantages in estimating the aleatoric uncertainty later.

[1] $\sigma_{\text{noise}}=\sigma_{\text{train}}=\sigma_{\text{test}}$

[2] $\boldsymbol{\Sigma}$ is a diagonal matrix and the elements on the diagonal are the singular values of $\boldsymbol{\Sigma}_{\text{train}}^{-1}\boldsymbol{Z}(\boldsymbol{X})$; $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal; a detailed derivation is presented in Appendix.

[3] Eq. 8b is also in the format of a summation for a quadratic term of $\boldsymbol{z}(\boldsymbol{x}_*)$ and a noise term.

Results

We apply the unmodified GPR (Eq. 5) on several synthetic datasets where both input x and output y are one dimensional to facilitate visualization. Although the presented results are based on one dimensional input x, they are generalizable to any dimensional input. The selected kernels are linear and RBF kernels, and we present the results of two scenarios with known and unknown noise levels.[4]

Dataset

Four synthetic datasets are generated and plotted in Fig. 2, and each of them contains 1000 points with a noise level of $\sigma_{\text{noise}} = 0.05$. Four other undersampled datasets are plotted in Fig. 3, each of which contains $1000 \times 5\% = 50$ points.

Fig 2. Original Datasets.

Fig 3. Undersampled Datasets.

In Dataset 1, both input $\boldsymbol{X}$ and output $\boldsymbol{y}$ follow a Gaussian distribution $\mathcal{N}(0,1^2)$ and are correlated with a Pearson coefficient of 0.75. Dataset 2 is transformed from Dataset 1, which moves the set of points where $x \geq 0$ in Dataset 1 along the line $y = x$ until the maximum input in that set equals 0, and moves the remaining points where $x < 0$ in Dataset 1 until the minimum input is 0. Dataset 3 has input $\boldsymbol{X}$ and output $\boldsymbol{y}$ uniformly distributed over a half-open interval $[-\pi,\pi)$. Output $\boldsymbol{y}$ of Dataset 4 is obtained by multiplying a factor function over output $\boldsymbol{y}$ from Dataset 3, which is defined as $f(\boldsymbol{x}) = \sin(\boldsymbol{x})/2{+}1$ and $\boldsymbol{x}$ is the corresponding input. We should note that the inputs $\boldsymbol{X}$ of original Datasets 3-4 are exactly same as shown in Fig. 2C-2D, and the inputs $\boldsymbol{X}$ of corresponding undersampled Datasets 3-4 are also identical as shown in Fig. 3C-3D.

GPR with Known Noise Level

Linear Kernel

The regression surface of GPR with linear kernel is a hyperplane and the variance is a quadric hypersurface defined by Eq. 9 in feature/original spaces, where the hyperplane always passes the origin, the variance is a function only with respect to the coordinate of testing input $\boldsymbol{x}_*$ and a unique minimum is located at $\boldsymbol{x}_* = \boldsymbol{0}$. Figs. 4-5 present results for GPR with linear kernel on the one dimensional synthetic datasets, where top sub-figures plot the reference models/predictions (red lines) overlapped on the data (blue dots), middle sub-figures show the derived variances (blue curves) across the original input space, and the bottom sub-figures shows the corresponding "z-score" for training set which is computed via Eq. 11 if the residual $(y{-}y_{\text{reference}})$ is mistakenly normalized by standard deviation Σ directly from GPR (Eq. 5b).

Fig 4. GPR with Linear Kernel on Original Datasets.

Fig 5. GPR with Linear Kernel on Undersampled Datasets.

[4] These results are produced by Python 3.8.7 and scikit-learn 0.24.0; although we have not tested, it should work with other version of Python and package as well. The code for this paper is available at: `https://github.com/nidaye1999/normative-model-GPR`.

$$z\text{-score} = \frac{y - y_{\text{reference}}}{\Sigma} \tag{11}$$

The mapping function of linear kernel projects an input to itself (Eq. 6b), and for one dimensional input, Eq. 9 can be further reduced to[5]

$$\begin{aligned}\Sigma_*^2(x_*) &= \left[1 - \boldsymbol{\Sigma}^T \left(\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T + \boldsymbol{I}\right)^{-1} \boldsymbol{\Sigma}\right] x_*^2 + \sigma_{\text{test}}^2 \\ &= \left(1 - \frac{\sigma^2}{\sigma^2+1}\right) x_*^2 + \sigma_{\text{test}}^2 \\ &= \frac{x_*^2}{\sigma^2+1} + \sigma_{\text{test}}^2,\end{aligned} \tag{12}$$

where σ is a scalar and equal to the only one singular value of $\boldsymbol{\Sigma}_{\text{train}}^{-1}\boldsymbol{X}$. As shown in Figs. 4-5, the variance is a univariate function of coordinate of the testing input x_* where the shape is a quadratic curve, and the global minimum is always located at $x_* = 0$ with a value of $\sigma_{\text{test}}^2 = 0.05^2$ as Eq. 12 formulated. The result of GPR with linear kernel presents a good example which illustrates the derived variance from GPR does not model the conditional variance $Var(y|\boldsymbol{x})$, thus corresponding z-score cannot be utilized as a normalized deviation in a normative model.

As previously mentioned, the predicted variance for testing set from GPR only depends on the training set input and the kernel function. As the original as well as the undersampled Datasets 3-4 have identical inputs $\boldsymbol{X}$, the variance curves in Figs. 4C-4D and Figs. 5C-5D are respectively identical.

RBF Kernel

Unlike the linear kernel, RBF kernel mapping function (Eq. 7b) defines a feature space which is different from the original space. Therefore, regarding the original space, the regression surface is no longer a hyperplane and the variance is no more a quadric hypersurface for the RBF kernel.[6] Because the mapping function of RBF kernel is very complicated, we only briefly describe the characteristics of the regression surface and variance in the original space. For a test input $\boldsymbol{x}_*$, the prediction is a summation of discounted outputs of all training points where each corresponding discount factor is determined by the Euclidean distance between $\boldsymbol{x}_*$ and that training input, and the predicted value converges to 0 if $\boldsymbol{x}_*$ is far away from all training inputs. On the other hand, the variance depends only on the density of training inputs at $\boldsymbol{x}_*$, and higher density results in lower variance. Therefore, the variance of GPR with RBF kernel is related of the relative location to the training inputs rather than the absolute location specified by coordinate.

The results for GPR with RBF kernel applied to these synthetic datasets are shown in Figs. 6-7.[7] As shown in Figs. 6-7, the variance is unrelated to the conditional variance $Var(y|x)$. Therefore, z-scores based on this model do not represent normalized deviation. However, unlike the quadratic curves whose unique minimum is always located at $x = 0$ in Figs. 4-5 for linear kernel, the variance function of GPR with RBF kernel regarding the original input space is related to the distribution of training input $\boldsymbol{X}$. The denser inputs at the middle of Dataset 1 and two ends of Dataset 2 lead

[5] $z(x) = x$ is an one dimensional vector (scalar) in this example, $\boldsymbol{\Sigma}$ is a $m \times 1$ matrix where $\Sigma_{1,1}$ equals σ and the rest elements are 0s, $\boldsymbol{V}$ is an 1×1 matrix and the only one element equals 1.

[6] In the feature space, regression surface is always a hyperplane and variance is always a quadric hypersurface for any kernels.

[7] The value of hyper-parameter l in Eq. 7a does not affect the main idea of this paper, thus we used a fixed value of 1 instead of hyper-parameter optimization.

to lower variances at those locations in Figs. 6A-6B, while the uniformly distributed inputs of Datasets 3-4 result in relatively flat curves in Figs. 6C-6D. According to Eq. 7a and given an arbitrary input $\boldsymbol{x}_*$, the RBF kernel function returns a larger value for a point in $\boldsymbol{X}$ that is closer to $\boldsymbol{x}_*$, and $\boldsymbol{k}(\boldsymbol{x}_*,\boldsymbol{X})$ and $\boldsymbol{k}(\boldsymbol{X},\boldsymbol{x}_*)$ have more large elements if $\boldsymbol{x}_*$ is close to more points in $\boldsymbol{X}$. Both result in the decrease of the value for Eq. 8a, i.e., to smaller variance.[8]

Fig 6. GPR with RBF Kernel on Original Datasets.

Fig 7. GPR with RBF Kernel on Undersampled Datasets. Scales of Y-axis for variance plots are different.

Similar to the result for the linear kernel, the theoretical minimum of variance is $\sigma^2_{\text{test}} = 0.05^2$,[9] and the variance curves are exactly identical in Figs. 6C-6D and Figs. 7C-7D respectively.

GPR with Unknown Noise Level

The noise level can be included as a hyper-parameter when it is unknown. However, the derived variance from GPR still does not model the heterogeneity $Var(y|\boldsymbol{x})$, although it could be a good approximation in some special cases.

As the basic properties of linear and RBF kernels have been introduced, a hybrid kernel is utilized in the following analysis which is defined as

$$k_{\text{hybrid}}(\cdot,\cdot) = w_{\text{linear}} k_{\text{linear}}(\cdot,\cdot) + w_{\text{RBF}} k_{\text{RBF}}(\cdot,\cdot) + k_{\text{white}}(\cdot,\cdot), \tag{13}$$

where w_{linear} and w_{RBF} represent adjustable weights on linear and RBF kernels, and $k_{\text{white}}(\cdot,\cdot)$ refers to a white-noise kernel that represents the $\boldsymbol{\Sigma}^2_{\text{noise}}$.[10] The Eq. 5b is reformulated as

$$\boldsymbol{\Sigma}^2_* = \boldsymbol{K}_{\text{hybrid}}(\boldsymbol{X}_*,\boldsymbol{X}_*) - \boldsymbol{K}_{\text{hybrid}}(\boldsymbol{X}_*,\boldsymbol{X}) \boldsymbol{K}_{\text{hybrid}}(\boldsymbol{X},\boldsymbol{X})^{-1} \boldsymbol{K}_{\text{hybrid}}(\boldsymbol{X},\boldsymbol{X}_*). \tag{14}$$

Original Datasets 3-4 in Figs. 2C-2D are prefect for testing whether a model captures the heterogeneity $Var(y|\boldsymbol{x})$, as the large number of instances and uniformly distributed data over the input space lead to negligible epistemic uncertainty in certain input range, and the true reference model $y = 0$ is very simple.[11] The hyper-parameters are tuned by maximizing the likelihood $P(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents all hyper-parameters in the model. The results are plotted in Fig. 8, and the optimized hyper-parameters are listed in Table 1 as well as the overall variances of residual $Var(y - y_{\text{reference}})$.

Fig 8. GPR with Hybrid Kernel on Original Datasets 3-4. Scales of Y-axis for variance plots are different.

As shown in Fig. 8, the GPR accurately estimates the reference models, i.e., $y_{\text{reference}} \approx y_{\text{reference,true}}$. The variance curves are nearly quadratic, since the w_{linear} is relatively larger than w_{RBF} while w_{RBF} is not exact zero as listed in Table 1. However, the domination of $k_{\text{white}}(\cdot,\cdot)$ over $k_{\text{linear}}(\cdot,\cdot)$ and $k_{\text{RBF}}(\cdot,\cdot)$ due to small optimized weights flattens the curves, i.e., the value of the curve is almost constant over the

[8] $\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) + \boldsymbol{\Sigma}^2_{\text{train}}$ is a symmetric positive definite matrix.

[9] It is a theoretical lower limit, the smallest variance for a given dataset is very likely greater than σ^2_{test}.

[10] $\boldsymbol{K}_{\text{white}}(\boldsymbol{X},\boldsymbol{X}) = \boldsymbol{\Sigma}^2_{\text{train}}$, $\boldsymbol{K}_{\text{white}}(\boldsymbol{X}_*,\boldsymbol{X}_*) = \boldsymbol{\Sigma}^2_{\text{test}}$, $\boldsymbol{K}_{\text{white}}(\boldsymbol{X},\boldsymbol{X}_*) = \boldsymbol{0}$, and $\boldsymbol{K}_{\text{white}}(\boldsymbol{X}_*,\boldsymbol{X}) = \boldsymbol{0}$.

[11] Two datasets with quadratic reference models are presented in Appendix.

Table 1. Optimized Hyper-parameters for Hybrid Kernel on Original Datasets 3-4.

	w_{linear}	w_{RBF}	l	σ^2_{noise}	$Var(y{-}y_{\text{reference}})$
Dataset 3	7.29e-8	4.88e-10	2.94e2	3.29	3.29
Dataset 4	1.35e-7	1.98e-17	1.54e-5	3.64	3.64

plotted input range in this example. Particularly, the σ^2_{noise} is very close to the overall residual variance $Var(y{-}y_{\text{reference}})$, and the explanation will be presented later. Therefore, $\boldsymbol{K}_{\text{hybrid}}(\boldsymbol{X}_*,\boldsymbol{X}_*) \approx \boldsymbol{\Sigma}^2_{\text{test}}$, $\boldsymbol{K}_{\text{hybrid}}(\boldsymbol{X},\boldsymbol{X}) \approx \boldsymbol{\Sigma}^2_{\text{train}}$, $\boldsymbol{K}_{\text{hybrid}}(\boldsymbol{X}_*,\boldsymbol{X}) \approx \boldsymbol{0}$ and $\boldsymbol{K}_{\text{hybrid}}(\boldsymbol{X},\boldsymbol{X}_*) \approx \boldsymbol{0}$, which result in $\boldsymbol{\Sigma}^2_* \approx \boldsymbol{\Sigma}^2_{\text{test}} = \boldsymbol{\Sigma}^2_{\text{noise}}$.

Regarding Eq. 1, since the noise is included as a tunable hyper-parameter without any constraints, the optimizer will adjust reference model $f(\cdot)$ as well as bias σ^2_{noise} to $Var(y{-}y_{\text{reference}})$ to maximize the likelihood $P(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta})$. Even the σ_{noise} refers to the observation noise level in GPR while the optimizer handles it as a variable without considering its meaning in a model.

In Dataset 3, σ^2_{noise} is biased to the overall residual variance $Var(y{-}y_{\text{reference}})$, and $Var(y{-}y_{\text{reference}})$ is well matched with the homoskedastic heterogeneity $Var(y|\boldsymbol{x})$. So the z-scores plotted in Fig. 8A show the GPR works as a normative model approach in this special case. However, in Dataset 4, σ^2_{noise} is also biased to the overall residual variance $Var(y{-}y_{\text{reference}})$, while $Var(y{-}y_{\text{reference}})$ does not approximate the heteroskedastic heterogeneity $Var(y|\boldsymbol{x})$. So the z-scores plotted in Fig. 8B do not represent a measure of normalized deviation in general.

Discussion

Although GPR could be extended and to model the heterogeneity as presented in this work, it is either: (1) hard to estimate the aleatoric uncertainty accurately when the data are sparse, e.g., at the middle of Dataset 2; or (2) unnecessary to model the conditional variance by Eq. 10 when the data are dense, e.g., Datasets 3-4. One approach to estimate $\sigma^2_{\text{aleatoric}}(\boldsymbol{x}_*)$ is using the sliding window technique, but it is hard to choose the window size for each dimension of input. For Scenario 1, even if the optimal window sizes can be obtained, it is hard to accurately estimate $\sigma^2_{\text{aleatoric}}(\boldsymbol{x}_*)$ when the window centered at $\boldsymbol{x}_*$ only covers a small number of training data points, e.g., $\boldsymbol{x}_*$ is far away from all points in $\boldsymbol{X}$. If the window centered at $\boldsymbol{x}_*$ covers a large number of training data points, e.g., Scenario 2, $Var(y|\boldsymbol{x}_*)$ should almost equal $\sigma^2_{\text{aleatoric}}(\boldsymbol{x}_*)$ and epistemic uncertainty is insignificant. Then $Var(y|\boldsymbol{x}_*)$ can be simply approximated as a local variance over a space defined by the window.

Another misunderstanding we found in the literature is interpreting the noise term σ^2_{noise} as aleatoric uncertainty. When the observation noise is considered as a hyper-parameter, it will likely bias the overall residual variance $Var(y{-}y_{\text{reference}})$. The overall residual variance is a good approximation of homoskedastic aleatoric uncertainty $Var(y|\boldsymbol{x})$. It is, however, not valid for cases with heteroskedastic residuals, which is the main motivation for using normative modeling. Although the value of the noise term is biased to estimate overall residual variance during the optimization, the mathematical/physical meanings are pre-defined by the model. Moreover, in homoskedastic aleatoric uncertainty cases, further investigation is needed to verify whether $\boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X}_*){-}\boldsymbol{K}(\boldsymbol{X}_*,\boldsymbol{X})\left[\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}){+}\boldsymbol{\Sigma}^2_{\text{noise}}\right]^{-1}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}_*)$ will still be a good approximation of epistemic uncertainty with such a biased estimation of observation noise level.[12]

[12] $\boldsymbol{\Sigma}^2_{\text{noise}} \ll \boldsymbol{\Sigma}^2_{\text{aleatoric}}$, so $\boldsymbol{\Sigma}^2_{\text{noise}}{+}\boldsymbol{\Sigma}^2_{\text{aleatoric}} \approx \boldsymbol{\Sigma}^2_{\text{aleatoric}}$ in Eq. 10.

Conclusion

In this paper, we present the mathematical derivation with a general formula to demonstrate that the derived prediction variance from GPR does not model the heterogeneity $Var(y|\boldsymbol{x})$, which in general is necessary for a normative model. GPR with a linear kernel and an RBF kernel are used as examples to illustrate this statement on one dimensional input datasets. Overall, the derived variance from GPR cannot be utilized in a normative model alone.

Supporting information

S1 Appendix. This file contains Eq. S1, Fig. S1, and Table S1. Eq. S1, a detailed derivation for Eq. 9. Fig. S1 and Table S1, results for modified Datasets 3-4.

Acknowledgments

We would like to express our appreciation to the anonymous reviewers for insightful discussions and feedback that have improved our study and manuscript. Also, special thanks to Laureate Institute for Brain Research and the University of Tulsa for supporting this research.

References

1. Kuczmarski RJ. CDC growth charts: United States. 314. US Department of Health and Human Services, Centers for Disease Control and Prevention; 2000.

2. Kuczmarski RJ. 2000 CDC Growth Charts for the United States: methods and development. 246. US Department of Health and Human Services, Centers for Disease Control and Prevention; 2002.

3. World Health Organization. Reproduced from "Weight-for-Age Boys: Birth to 2 years (percentiles)"; accessed 6-May-2021; Copyright (2021). Available from: `https://cdn.who.int/media/docs/default-source/child-growth/child-growth-standards/indicators/weight-for-age/boys-charts---weight-for-age-birth-to-2-years-(percentiles).pdf`.

4. Wikipedia contributors. Uncertainty quantification — Wikipedia, The Free Encyclopedia; 2021. Available from: `https://en.wikipedia.org/w/index.php?title=Uncertainty_quantification&oldid=1015674163`.

5. Schulz E, Speekenbrink M, Krause A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. Journal of Mathematical Psychology. 2018;85:1–16. doi:https://doi.org/10.1016/j.jmp.2018.03.001.

6. Tonner PD, Darnell CL, Engelhardt BE, Schmid AK. Detecting differential growth of microbial populations with Gaussian process regression. Genome research. 2017;27(2):320–333.

7. Banerjee A, Dunson DB, Tokdar ST. Efficient Gaussian process regression for large datasets. Biometrika. 2013;100(1):75–89.

8. Raissi M, Babaee H, Karniadakis GE. Parametric Gaussian process regression for big data. Computational Mechanics. 2019;64(2):409–416.

9. Ziegler G, Ridgway GR, Dahnke R, Gaser C, Initiative ADN, et al. Individualized Gaussian process-based prediction and detection of local and global gray matter abnormalities in elderly subjects. NeuroImage. 2014;97:333–348.

10. Marquand AF, Rezek I, Buitelaar J, Beckmann CF. Understanding heterogeneity in clinical cohorts using normative models: beyond case-control studies. Biological psychiatry. 2016;80(7):552–561.

11. Wolfers T, Beckmann CF, Hoogman M, Buitelaar JK, Franke B, Marquand AF. Individual differences v. the average patient: mapping the heterogeneity in ADHD using normative models. Psychological Medicine. 2020;50(2):314–323.

12. Wolfers T, Doan NT, Kaufmann T, Alnæs D, Moberget T, Agartz I, et al. Mapping the heterogeneous phenotype of schizophrenia and bipolar disorder using normative models. JAMA psychiatry. 2018;75(11):1146–1155.

13. Zabihi M, Oldehinkel M, Wolfers T, Frouin V, Goyard D, Loth E, et al. Dissecting the heterogeneous cortical anatomy of autism spectrum disorder using normative models. Biological Psychiatry: Cognitive Neuroscience and Neuroimaging. 2019;4(6):567–578.

14. Rasmussen CE, Williams CKI. Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press; 2005.

15. Do CB. Gaussian processes. Stanford University. 2008;.

16. Mohri M, Rostamizadeh A, Talwalkar A. Foundations of machine learning. MIT press; 2018.

17. Shashua A. Introduction to Machine Learning: Class Notes 67577; 2009.

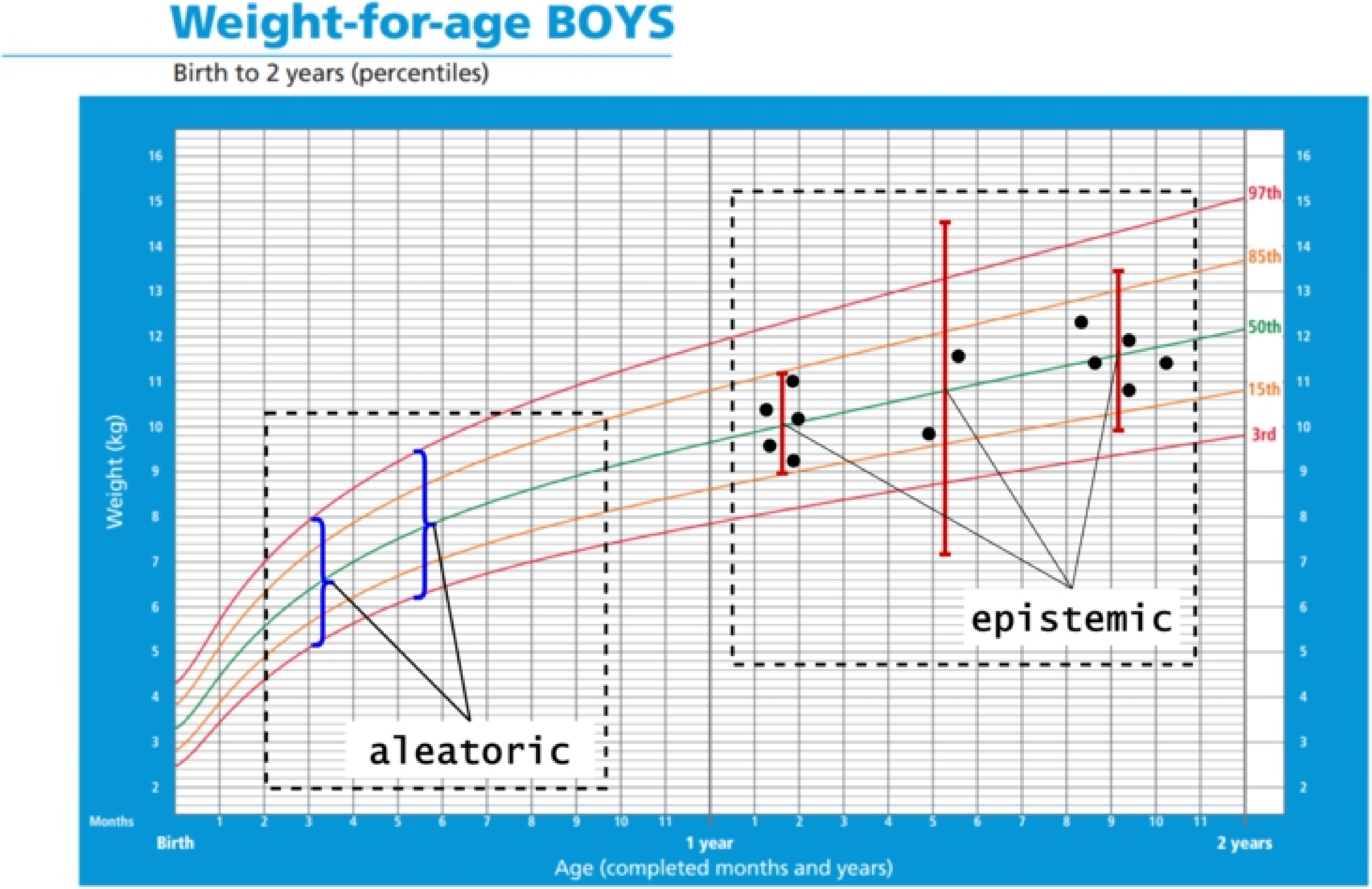
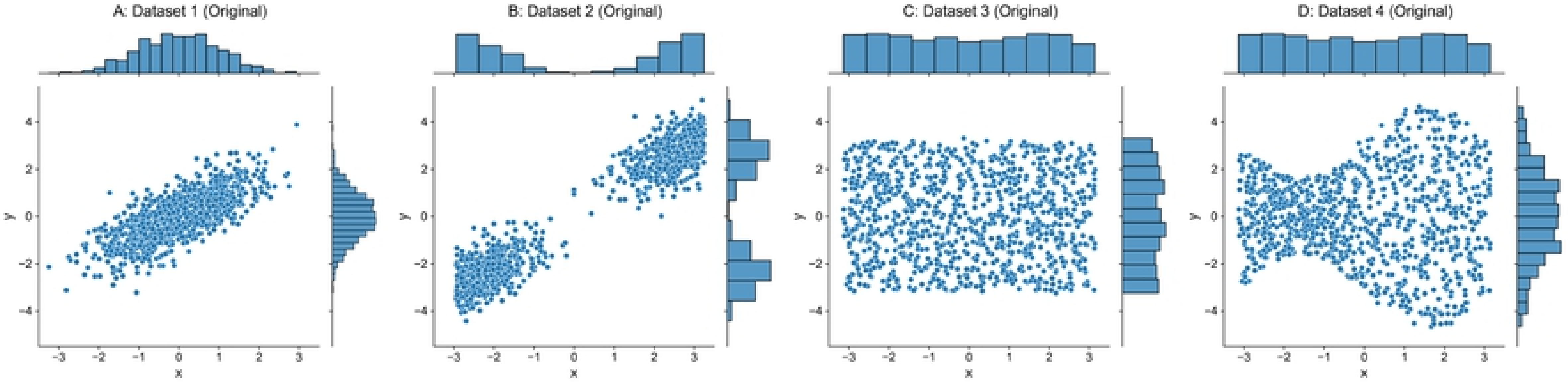

Weight-for-age BOYS
Birth to 2 years (percentiles)
Weight (kg)
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
97th
85th
50th
15th
3rd
Months
1
2
3
4
5
6
7
8
9
10
11
Birth
1 year
2 years
Age (completed months and years)
aleatoric
epistemic

fig1



A: Dataset 1 (Original)

B: Dataset 2 (Original)

C: Dataset 3 (Original)

D: Dataset 4 (Original)

fig2

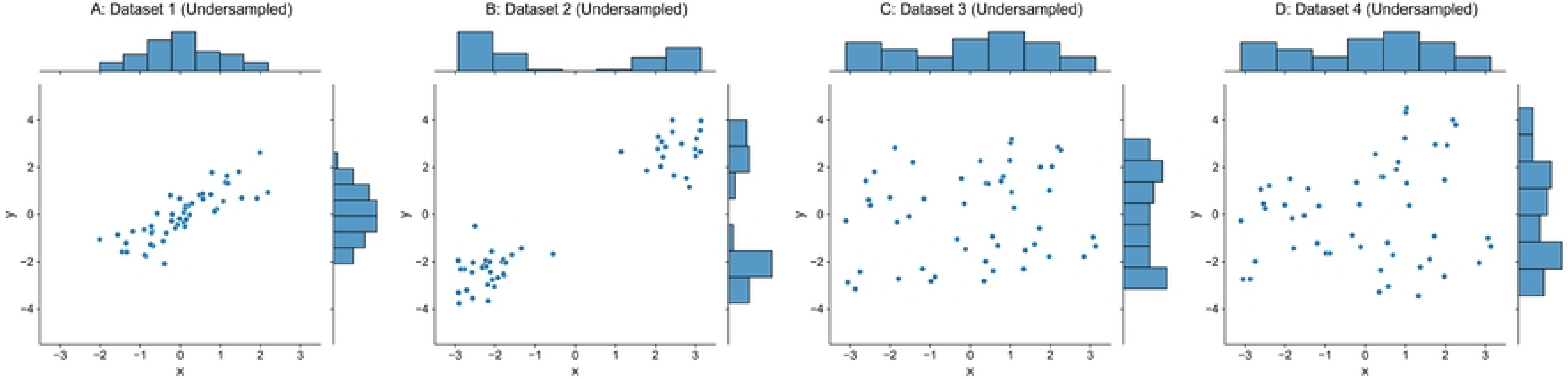

A: Dataset 1 (Undersampled)
B: Dataset 2 (Undersampled)
C: Dataset 3 (Undersampled)
D: Dataset 4 (Undersampled)
y
4
2
0
−2
−4
x
−3
−2
−1
0
1
2
3

fig3

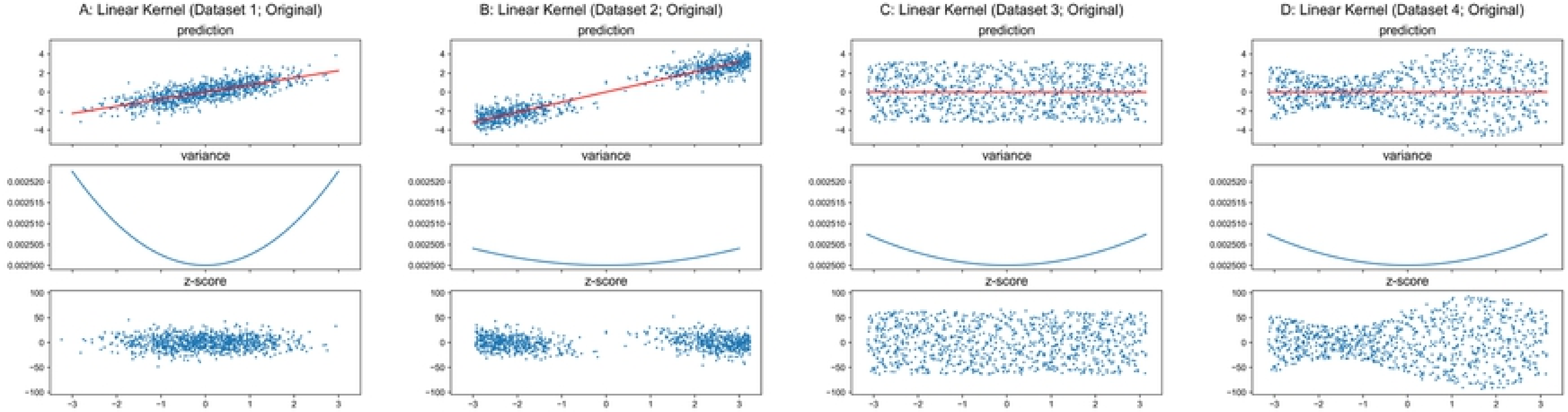

A: Linear Kernel (Dataset 1; Original)
B: Linear Kernel (Dataset 2; Original)
C: Linear Kernel (Dataset 3; Original)
D: Linear Kernel (Dataset 4; Original)
prediction
variance
z-score
4
2
0
-2
-4
0.002520
0.002515
0.002510
0.002505
0.002500
100
50
0
-50
-100
-3
-2
-1
0
1
2
3

fig4

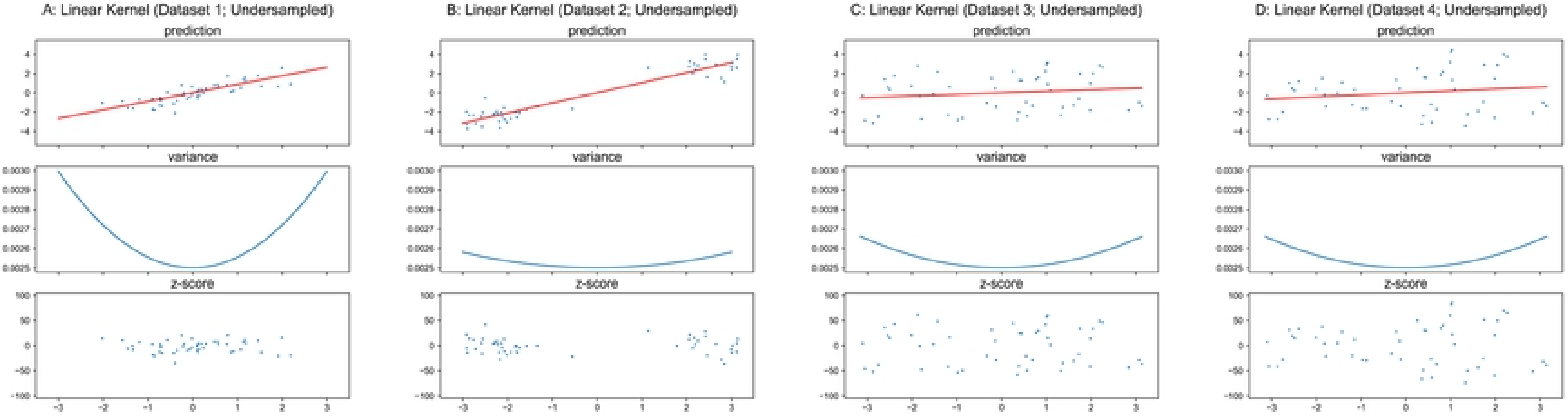
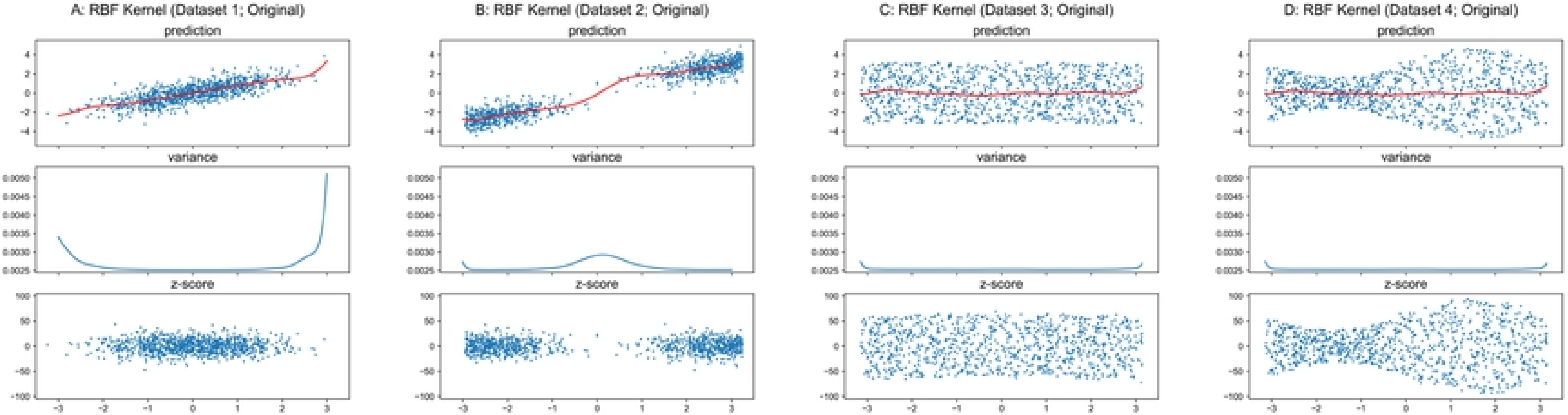

A: Linear Kernel (Dataset 1; Undersampled)
B: Linear Kernel (Dataset 2; Undersampled)
C: Linear Kernel (Dataset 3; Undersampled)
D: Linear Kernel (Dataset 4; Undersampled)
prediction
variance
z-score
4
2
0
-2
-4
0.0030
0.0029
0.0028
0.0027
0.0026
0.0025
100
50
0
-50
-100
-3
-2
-1
0
1
2
3

fig5



A: RBF Kernel (Dataset 1; Original)
B: RBF Kernel (Dataset 2; Original)
C: RBF Kernel (Dataset 3; Original)
D: RBF Kernel (Dataset 4; Original)
prediction
variance
z-score
4
2
0
-2
-4
0.0050
0.0045
0.0040
0.0035
0.0030
0.0025
100
50
0
-50
-100
-3
-2
-1
0
1
2
3

fig6

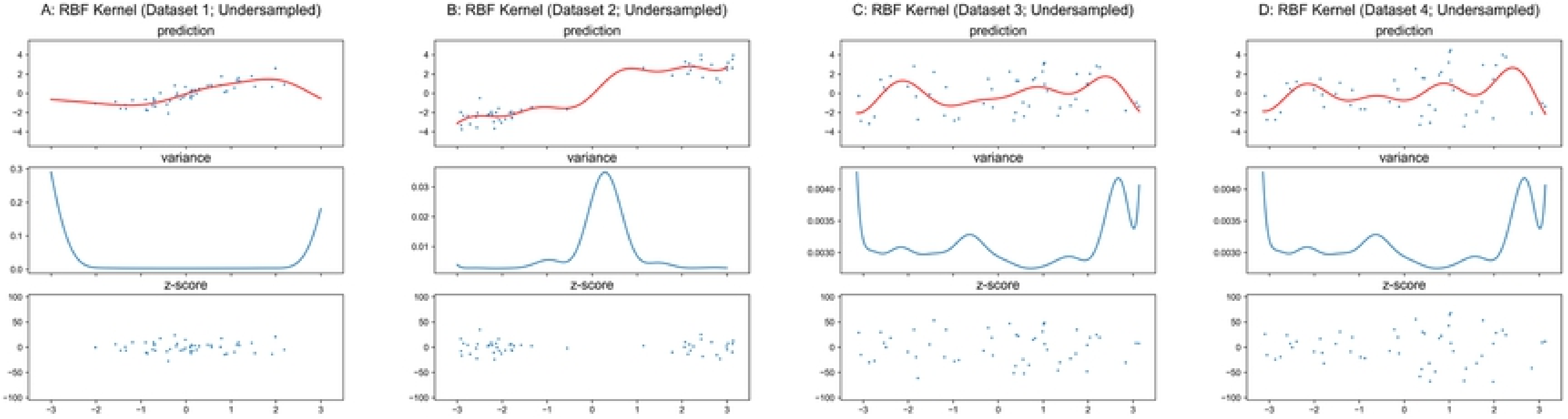

A: RBF Kernel (Dataset 1; Undersampled)
B: RBF Kernel (Dataset 2; Undersampled)
C: RBF Kernel (Dataset 3; Undersampled)
D: RBF Kernel (Dataset 4; Undersampled)
prediction
variance
z-score
4
2
0
-2
-4
0.3
0.2
0.1
0.0
0.03
0.02
0.01
0.0040
0.0035
0.0030
100
50
0
-50
-100
-3
-2
-1
0
1
2
3

fig7

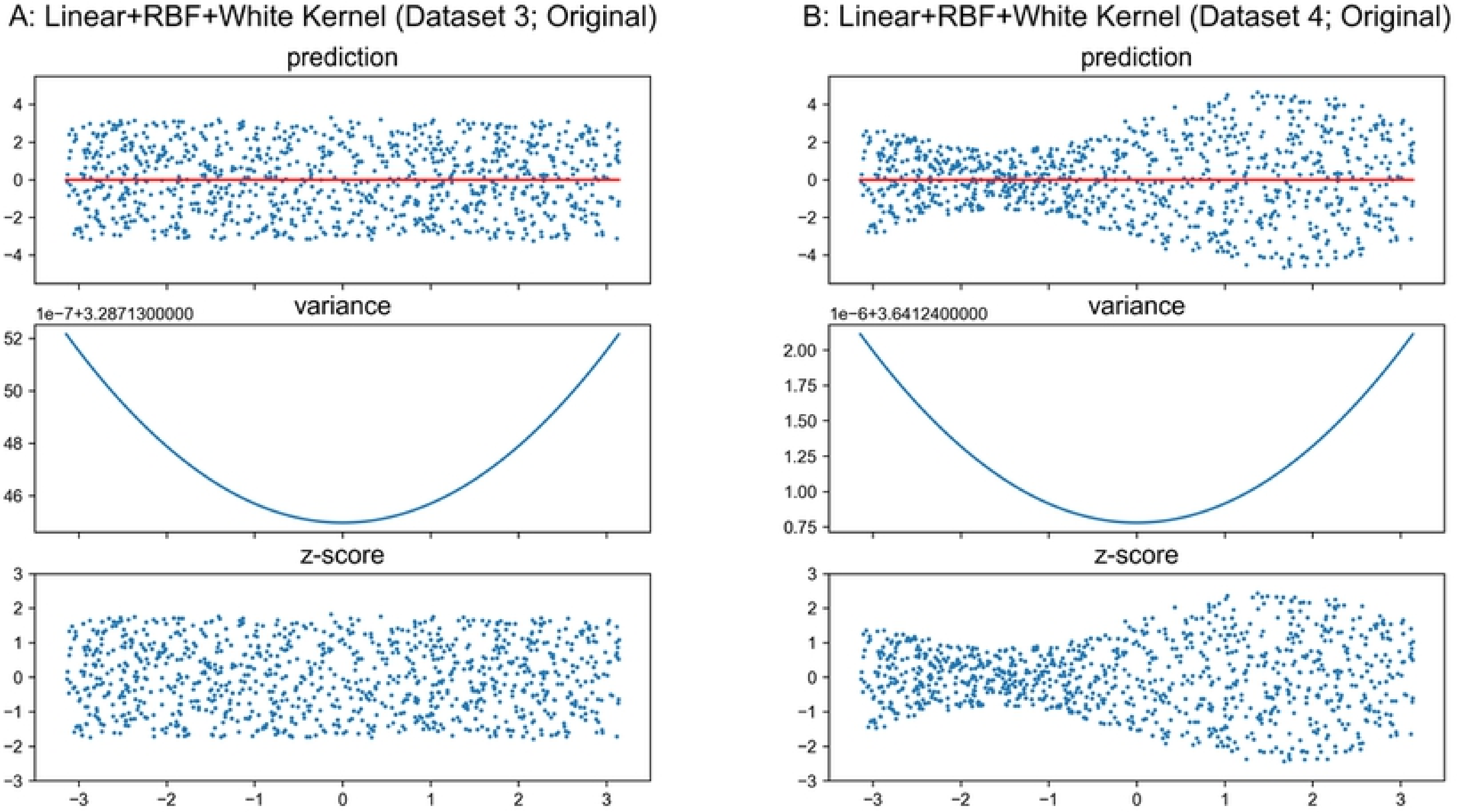

A: Linear+RBF+White Kernel (Dataset 3; Original)
B: Linear+RBF+White Kernel (Dataset 4; Original)
prediction
4
2
0
-2
-4
variance
1e-7+3.2871300000
52
50
48
46
1e-6+3.6412400000
2.00
1.75
1.50
1.25
1.00
0.75
z-score
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3

fig8