MULTI-FIDELITY MODEL

WITH DIMENSION REDUCTION

A Thesis

Submitted to the Faculty

of

Purdue University

by

Bangde Liu

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Mechanical Engineering

December 2020

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF THESIS APPROVAL

Dr. Guang Lin, Chair

School of Mechanical Engineering

Dr. Ilias Bilionis

School of Mechanical Engineering

Dr. Baijian Yang

School of Computer and Information Technology

Approved by:

Dr. Nicole Key

Head of the School Graduate Program

This is dedicated to my families, my advisors and my friends.

ACKNOWLEDGMENTS

Thanks to Professor Lin, Professor Ilias, and Professor Yang, I cannot finish the thesis without their professional suggestions and instructions. I also want to say thanks to my research group members. They provide me lots of advice when I writing the thesis.

During the period of my master studies, I have met many great professors and friends at Purdue and learned a lot from them. I feel grateful for their help. Last I want thanks to my beloved family for their loving consideration and great confidence in me through these years. When I face difficulties in life and studies, they make me regain courage and moving forward.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

SYMBOLS

$\boldsymbol{D}$	data-set
D	original data dimension
DR	dimension reduction
d, d_{DR}	the dimension that we want to get after dimension reduction
R	set of reals
$\boldsymbol{W}$	projection matrix
g	link function
$V_d(\cdot)$	Stiefel Manifold
∇	gradient information
$\boldsymbol{K}$	covariance matrix

ABBREVIATIONS

GP	Gaussian process
FD	Finite difference
AR1	Linear autoregressive multi-fidelity mode
NARGP	Nonlinear autoregressive multi-fidelity mode
L-BFGS	Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm
PCA	Principle component analysis
MSE	Mean square error
SVD	Singular value decomposition
KdV	Korteweg-de Vries
BIC	Bayesian information criterion
CDF	Cumulative distribution function
PDF	Probability density function

ABSTRACT

Liu, Bangde MS, Purdue University, December 2020. Multi-fidelity model with dimension reduction. Major Professor: Guang L. Professor.

In scientific and engineering applications, often sufficient low-cost low-fidelity data is available while only a small fractional of high-fidelity data is accessible. The multi-fidelity model integrates a large set of low-cost but biased low-fidelity datasets with a small set of precise but high-cost high-fidelity data to make an accurate inference of quantities of interest. Under many circumstances, the number of model input dimensions is often high in real applications. To simplify the model, dimension reduction is often used. The gradient-free active subspace is employed in this research for dimension reduction. In this work, we build a predictive model for high-dimensional nonlinear problems by integrating the nonlinear multi-fidelity Gaussian progress regression and the gradient-free active subspace method. Numerical results demonstrated that the proposed approach can not only perform effective dimension reduction on the original data but also obtain accurate prediction results thanks to the effective dimension reduction procedure.

1. INTRODUCTION

The main idea of the multi-fidelity models [1–5] can be constructed by combining computational-expensive high accurate (high-fidelity) data with low-cost less accurate data (low-fidelity). Nowadays, most of the multi-fidelity approaches are set up on the Gaussian process (GP) [6] with the order-one autoregressive model proposed by [7]. GP [8–15] is a suitable method for the multi-fidelity problems, because it has ability to use the prior belief to learn how different fidelities are related. However, the traditional autoregressive multi-fidelity models are only suitable when different fidelities relationship is linear. To cope with the nonlinear relation between fidelities, a method called nonlinear information fusion algorithm [16] that is based on the GP and nonlinear autoregressive scheme had been developed. This method proposes a more general form of the multi-fidelity model structure. Thanks to this method, It can deal with both the linear and nonlinear multi-fidelity problems effectively.

In the scientific and engineering applications, people use computer models to study the input and output relations. In most cases, the dimension of the input space is usually very large that makes the models to be computational expensive. Reducing dimensions can help otherwise infeasible parameter studies. To enable such studies, people usually use various methods to decrease the dimensions of the input space. There have many popular dimension reduction methods, such as principal component analysis (PCA) [17, 18], forward feature selection, backward feature elimination, and gradient-free active subspace approach [19]. Active subspace is a dimension reduction tool to identify the important directions of the input space. However, the classic active subspace method is based on the gradient information. In most cases, it is hard to get the gradient information. To avoid this shortcoming, a gradient-free active subspace method [19] is proposed.

In this work, a novel nonlinear multi-fidelity predictive model is built for high-dimensional problems that is based on the nonlinear multi-fidelity scheme and gradient-free active subspace method. In particular, the low-fidelity data can be used to calculate the BIC score to determine the active subspace dimensions that is used as the input of the gradient-free active subspace method to get the dimension reduction matrix. To improve the predictive accuracy, Bayesian active learning method [20] is employed to augment our original data based on the high-fidelity data to enhance accuracy. Bayesian active learning method indicates where a function will be evaluated next under a limited budget. In the proposed model, these new samples are explored according to where the largest variance of function is located. By using largest variance as the sample location indicator, new samples are added using Bayesian active learning to augment our original data size. Then the dimension reduction matrix is employed to perform the dimension reduction on our new low- and high-fidelity data. Finally, we use the data after dimension reduction as the input of our nonlinear multi-fidelity scheme to build the nonlinear multi-fidelity predictive model.

Our contribution in this paper is two-fold:

1. We propose a new surrogate model that can perform gradient-free dimension reduction to the original data and make an accurate model prediction based on data after dimension reduction.

2. We implement our proposed model and make a systematic comparison of the proposed model with the standard multi-fidelity model on nonlinear high-dimensional problems. We show the feasibility of our method by running some examples. Besides, we compare the new proposed model with the standard multi-fidelity model to show its advantage. The structure of this work is organized as follows: in Section 2, we introduce basic definitions of Gaussian process regression, gradient-free active subspace, nonlinear autoregressive algorithm, and Bayesian active learning. The main algorithm of our method is presented in Section 2.5, and numerical results are shown in Section 3. Summary and discussion are presented in Section 4.

2. METHODOLOGY

In practice, the number of dimensions of our inputs is relatively high, which makes the computation costly. To relieve the computational cost, low-fidelity data is first employed as the training data for dimension reduction, then the projected data is applied for the multi-fidelity work. Moreover, to get accurate results, Bayesian active learning method is employed to add additional samples to augment our original training data size for the multi-fidelity model. The goal of this work is to construct a nonlinear multi-fidelity predictive model for high-dimensional problems that can perform dimension reduction based on original data and make accurate prediction according to the projected data after dimension reduction. In conclusion, the new proposed model consists of Gaussian process regression, gradient-free active subspace method, multi-fidelity model, and Bayesian active learning approaches.

2.1 Gaussian Process Regression

GP regression is known as a nonparametric method. It is a supervised machine learning method. We assume our observation dataset as $\boldsymbol{D} = \{\boldsymbol{x}^i, \boldsymbol{y}^i\} = (\boldsymbol{x}, \boldsymbol{y})$ of $i = 1, \ldots, N$, and have:

$$\boldsymbol{y} = f(\boldsymbol{x}) \tag{2.1}$$

Where $f(\cdot)$ is the response surface, and $\boldsymbol{x} \in R^{D \times N}$.

GP regression calculates all the possible probability distribution over functions rather than the parameters of the specific function. The following steps are performed for GP regression:

First, a prior with zero mean is assigned to the response surface $f(\cdot)$. For example, $f \sim GP\left(f|\mathbf{0}, k\left(\boldsymbol{x}, \boldsymbol{x}'; \theta\right)\right)$. Here k is known as the kernel function or a covariance

matrix in the form $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j; \theta), \boldsymbol{K} \in R^{N \times N}$ and is defined positive symmetric. θ is the hyperparameters in the kernel function. Using Bayes rule, we can get the posterior GP by combining the prior belief with observation.

Hyperparameters θ can be obtained by maximizing the model's log-likelihood function:

$$\log p(\boldsymbol{y}|\boldsymbol{x}, \theta) = -\frac{1}{2}\log|\boldsymbol{K}| - \frac{1}{2}\boldsymbol{y}^T\boldsymbol{K}^{-1}\boldsymbol{y} - \frac{N}{2}\log 2\pi \tag{2.2}$$

Using the knowledge of the Bayes rule, the posterior distribution $p(\boldsymbol{f}|\boldsymbol{y}, \boldsymbol{x})$ and the prediction of the new output f_* with new input $\boldsymbol{x}_*$ can be got:

$$p(f_*|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{x}_*) = \mathcal{N}(f_*|\mu_*, \Sigma_*) \tag{2.3}$$

$$\mu_* = \boldsymbol{k}_*\boldsymbol{K}^{-1}\boldsymbol{y} \tag{2.4}$$

$$\Sigma_* = \boldsymbol{k}_{**} - \boldsymbol{k}_*\boldsymbol{K}^{-1}\boldsymbol{k}_*^T \tag{2.5}$$

Where $\boldsymbol{k}_* = k(\boldsymbol{x}, \boldsymbol{x}_*)$ and $\boldsymbol{k}_{**} = k(\boldsymbol{x}_*, \boldsymbol{x}_*)$. Prediction can be obtained using posterior mean μ_*.

2.2 Active Subspace

2.2.1 Classic active subspace approach

In this section, the classic active subspace method which is based on the gradient information [21–28] has been introduced. Let f be the multivariate response surface with $D \gg 1$. By putting input $\boldsymbol{x} \in R^{D \times N}$ into $f(\cdot)$, the output is $f(\boldsymbol{x})$, assume we have N input points and our measured points:

$$\boldsymbol{x} = \left\{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\right\} \tag{2.6}$$

$$\boldsymbol{y} = \left\{\boldsymbol{y}^1, \ldots, \boldsymbol{y}^N\right\} \tag{2.7}$$

Working in the high dimension regime, it is hard to discover and exploit the structure of the $f(\boldsymbol{x})$ unless it has some special structure. In this work, the response surface can be approximated in the following form:

$$f(\boldsymbol{x}) \approx g\left(\boldsymbol{W}^T\boldsymbol{x}\right) \tag{2.8}$$

Here the matrix $\boldsymbol{W} \in R^{D\times d}$ is called projected matrix. It projects the high dimension space R^D to the low dimension space R^d (active subspace), and g is the link function. Denote that the form of Eq. (2.8) can be described in other ways to make the response of the columns of $\boldsymbol{W}$ correspond to the directions of the input space that is the most sensitive. Mathematically, $\boldsymbol{W} \in V_d\left(R^D\right)$, where $V_d\left(R^D\right)$ is the $D \times d$ matrix with orthonormal columns,

$$V_d\left(R^D\right) := \left\{\boldsymbol{A} \in R^{D\times d} : \boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}_d\right\} \tag{2.9}$$

with $\boldsymbol{I}_d$ is the $d \times d$ unit matrix. $V_d\left(R^D\right)$ is the Stiefel manifold.

Here we review the classic active subspace method to find out the active subspace according to the gradient information. To deal with the response surface in high dimension, Eq. (2.8) is introduced. The classic approach finds the active subspace in the following two steps. First, it uses the gradient information to get the projection matrix $\boldsymbol{W}$. Secondly, all inputs are projected to the active subspace, and the map between the projected inputs and the output can be learned by using the GP regression. Since this method needs gradient information, assume its gradient of $f(\cdot)$ at each input point:

$$\boldsymbol{g} = \left\{\boldsymbol{g}^1, \ldots, \boldsymbol{g}^N\right\} \tag{2.10}$$

Where $\boldsymbol{g} = \nabla f\left(\boldsymbol{x}\right) \in R^{D\times N}$, and $\nabla f(\cdot)$ is the gradient of $f(\cdot)$

$$\nabla f(\cdot) = \left(\frac{\partial f(\cdot)}{\partial x_1}, \ldots, \frac{\partial f(\cdot)}{\partial x_D} \right) \tag{2.11}$$

(1) Finding out active subspace using gradient information

First, the matrix $\boldsymbol{C}$ can be defined:

$$\boldsymbol{C} := \int (\nabla f(\boldsymbol{x}))(\nabla f(\boldsymbol{x}))^T \rho(\boldsymbol{x}) d\boldsymbol{x} \tag{2.12}$$

Where the $\rho(\boldsymbol{x})$ is the PDF of the input space, since $\boldsymbol{C}$ is symmetric positive definite, it can be written as:

$$\boldsymbol{C} = \boldsymbol{V} \boldsymbol{\Lambda} \boldsymbol{V}^T \tag{2.13}$$

Where $\boldsymbol{\Lambda} = \text{diag}\,(\lambda_1, \ldots, \lambda_D)$ is a diagonal matrix, λ_i is the eigenvalue of $\boldsymbol{C}$, $\lambda_1 \geq \cdots \geq \lambda_D \geq 0$, and $\boldsymbol{V} \in R^{D \times D}$ is an orthogonal matrix. By choosing d, the largest eigenvalue from the whole eigenvalue, we can get:

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda_1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Lambda_2} \end{bmatrix} \tag{2.14}$$

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{V_1} & \boldsymbol{V_2} \end{bmatrix} \tag{2.15}$$

Where $\boldsymbol{\Lambda_1} = \text{diag}\,(\lambda_1, \ldots, \lambda_d)$, $\boldsymbol{V_1} = [v_{11}, \ldots, v_{1d}]$ and $\boldsymbol{\Lambda_2}$, $\boldsymbol{V_2}$ are defined similarly. Our projection matrix is $\boldsymbol{W} = \boldsymbol{V_1}$.

Since it is hard to calculate Eq. (2.12) directly, Monte Carlo method [29–31] can be used to approximate the integral. Assuming that the observed inputs are drawn from $\rho(\boldsymbol{x})$, the observed gradients can be used to approximates $\boldsymbol{C}$, see Eq. (2.10), by:

$$\boldsymbol{C_N} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{g}^i \left(\boldsymbol{g}^i \right)^T \tag{2.16}$$

To get the eigenvalue and eigenvector of $\boldsymbol{C_N}$, the singular value decomposition (SVD) [32] method can be applied. The dimensionality d can be determined by looking for the spot with sharp changes in the value of $\boldsymbol{C_N}$.

(2) Exploring the correlation between projected inputs and output

After the projection matrix $\boldsymbol{W}$ had been calculated, the projected inputs $\boldsymbol{z} \in R^{d\times N}$ can be got in the following form:

$$\boldsymbol{z} = \left\{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^N\right\} \tag{2.17}$$

Where $\boldsymbol{z} = \boldsymbol{W}^T\boldsymbol{x}$, the link function $g(\cdot)$ that connects the projected inputs to the output, see Eq. (2.8), it can be identified by using GP regression.

2.2.2 Gradient free active subspace method

In the previous part, the classic approach to find the active subspace had been discussed. To get the active subspace using the classic method, the gradient information is necessary. In practice, collect gradient information is a quiet challenge. In most cases, the gradient information can be obtained through the finite-difference method (FDM). However, this method is computational expensive. What's more, the approximate gradients can be gained by approximating models learning from the data. In general, the problem is a black-box problem and it is difficult to get the gradient information, not to speak to calculate the active subspace. To eliminate this limitation, [19] introduced a new method to obtain the active subspace without the gradient information. This new method can be achieved in the following two steps:

(a) In the GP regression, we assign a GP prior to using the mean and covariance function. Then a new covariance matrix can be defined and make the active subspace $\boldsymbol{W}$ as hyperparameters which can be learned from the data. the following GP kernel form had beed proposed to express the prior knowledge about the active subspace structure that is shown in Eq. (2.8):

$$k_{AS}\left(\boldsymbol{x}, \boldsymbol{x}'\right) = k_{\text{base}}\left(\boldsymbol{W}^T\boldsymbol{x}, \boldsymbol{W}^T\boldsymbol{x}'\right) \tag{2.18}$$

Where the $k_{\text{base}}(\cdot,\cdot)$ is the standard covariance matrix, for instance, Matern or Radial basis function (RBF) kernel. The newly defined kernel is used to express the

prior knowledge about the link function $g(\cdot)$. If our new active subspace kernel $k_{AS}(\cdot,\cdot)$ is identified, the hyperparameters can be calculated by getting the maximization of the log marginal likelihood function below:

$$\boldsymbol{W}, \theta, \sigma = \operatorname{argmax} \log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{W}, \theta, \sigma) \tag{2.19}$$

Where θ is the hyperparameters of the base kernel, and σ is the standard deviation of the likelihood noise.

(b) We need to enforce the positivity constraint on all hyperparameters. It is not difficult to set positive constraints to the hyperparameters $(\theta, \sigma) = \boldsymbol{\psi}$. However, it is quite challenging and important to strengthen the orthogonality constraints on the dimension reduction matrix $\boldsymbol{W}$. This goal is achieved by applying the complete methodology of paper [19] which introduces the iterative two-step likelihood maximization scheme. This scheme keeps $\boldsymbol{W}$ as a constant when optimizing the hyperparameter $\boldsymbol{\psi}$ and vice versa. We use the L-BFGS algorithm [33] to optimize the hyperparameter $\boldsymbol{\psi}$, and apply a adaptive gradient-ascent method on the Stiefel manifold [34] when the projection matrix $\boldsymbol{W}$ is optimized.

2.2.3 Determine the active subspace dimension

Here we will show how to identify the optimal active subspace dimension. Bayesian model selection provides us a possible solution. The rules of probability theory are employed to select among different hypotheses. The Bayesian model selection using the Bayesian information criterion (BIC). BIC score is one of the Bayesian criteria used for Bayesian model selection and it tends to be one of the most popular criteria. The BIC score of the d-dimensional model is:

$$\mathrm{BIC}_d = \mathcal{L}\left(\boldsymbol{\theta}_d; \boldsymbol{x}, \boldsymbol{y}\right) - \frac{1}{2} \#\left(\boldsymbol{\theta}_d\right) \log N \tag{2.20}$$

Where $\boldsymbol{\theta}_d = (\boldsymbol{W}_d, \boldsymbol{\psi}_d)$, $(\boldsymbol{x}, \boldsymbol{y})$ is the observation, $\mathcal{L}$ is the log-likelihood function. N is the number of observations, and $\#\boldsymbol{\theta}_d$ is the number of estimated parameters $\boldsymbol{\theta}_d$

which is consist of the number of the active dimension matrix $\#\boldsymbol{W}_d$ and the number of the hyper-parameter $\#\boldsymbol{\psi}_d$:

$$\#\boldsymbol{\theta}_d = \#\boldsymbol{W}_d + \#\boldsymbol{\psi}_d = dD + \#\boldsymbol{\psi}_d \tag{2.21}$$

In Equation 2.20, the BIC is equal to the maximum log-likelihood function minus a term $\frac{1}{2}\#\left(\boldsymbol{\theta}_d\right)\log N$ which is used to penalize the model complexity. Now we will show how to determine the optimal number of active subspace dimensions. Assume the original dimension of model M_D is D, then the gradient-free active subspace method can be help us to get the model at d dimension which can be denoted as M_d, where $D \geq d$. Assume $BIC_{M_{d+1}} - BIC_{M_d} = \Delta$. If Δ>0, we draw the conclusion that M_d is better than M_{d+1}. If the Δ>5, the evidence is stronger. In other words, if there is a sharp increase of BIC from d to $d+1$, M_d is closer to the truth. At this point, the optimal dimension of the active subspace is d.

2.3 Multi-fidelity Modeling

2.3.1 Linear autoregressive multi-fidelity model (AR1)

The main scheme of the multi-fidelity method is GP regression and autoregressive scheme. GP regression is a non-parametric regression model to construct the probability model that enables the combination of different fidelity information sources. Assuming there have s levels of information sources, the inputs and output are denoted as $\boldsymbol{x}_q$ and $\boldsymbol{y}_q$, then the datasets at different levels of fidelity can be denoted as $\boldsymbol{D}_q = \left\{\boldsymbol{x}_q, \boldsymbol{y}_q\right\}, q = 1, \ldots, s$. Whereas $\boldsymbol{y}_s$ denotes the most accurate but computational expensive output, $\boldsymbol{y}_1$ is the low-cost but least accurate output. After these settings, the linear autoregressive scheme [7, 35] had been introduced:

$$f_q(\boldsymbol{x}) = \rho f_{q-1}(\boldsymbol{x}) + \delta_q(\boldsymbol{x}) \tag{2.22}$$

Where f_q and f_{q-1} are the GP models at the fidelity levels q and $q-1$, respectively. ρ is just a constant which measures the correlation between the model outputs $\left\{\boldsymbol{y}_q, \boldsymbol{y}_{q-1}\right\}$. And $\delta_q\left(\boldsymbol{x}_q\right)$ follows the Gaussian distribution, and its mean μ_{δ_q} and covariance function k_q. It can be written as $\delta_q \sim GP\left(\delta_q|\mu_{\delta_q}, k_q\left(\boldsymbol{x}_q, \boldsymbol{x}_q'; \theta_q\right)\right)$.

The linear autoregressive multi-fidelity scheme can be constructed by introducing the idea put forward by paper [35]. The main idea of the their method is to replace the GP prior $f_{q-1}(\boldsymbol{x})$ with the lower fidelity's GP posterior $f_{*_{q-1}}(\boldsymbol{x})$, and assume that the our datasets $\{D_1, D_2, \ldots, D_s\}$ are a nested structure datasets. It can be written as $D_1 \subseteq D_2 \subseteq D_s$. Generally speaking, this statement means that our high-fidelity level's training inputs are the subset of the low-fidelity levels training inputs. According to [35], this structure can be treated the same way as Gaussian posterior prediction by [36]. At this point, this problem is just a GP regression problem. The Gaussian posterior distribution at different fidelity $p\left(\boldsymbol{f}_q|\boldsymbol{y}_q, \boldsymbol{x}_q, f_{*_{q-1}}\right), q = 1, \ldots, s$ can be obtained, and the predictive mean and variance can be written in the following form:

$$\mu_{*q} = \rho\mu_{*q-1} + \mu_{\delta_q} + \boldsymbol{k}_{*q}\boldsymbol{K}_q^{-1}[\boldsymbol{y}_q - \rho\mu_{*q-1}\left(\boldsymbol{x}_q\right) - \mu_{\delta_q}] \tag{2.23}$$

$$\Sigma_{*q} = \rho^2\Sigma_{*q-1} + \boldsymbol{k}_{**} - \boldsymbol{k}_{*q}\boldsymbol{K}_q^{-1}\boldsymbol{k}_{*q}^T \tag{2.24}$$

2.3.2 Nonlinear autoregressive multi-fidelity model (NARGP)

However, the GP basis of the autoregressive model is suitable when the mapping between fidelities is linear. This model is not working when the mapping between fidelities is non-linear. The model proposed in [37–40] can be employed as follows:

$$f_q(\boldsymbol{x}) = \rho_q\left(f_{q-1}(\boldsymbol{x})\right) + \delta_q(\boldsymbol{x}) \tag{2.25}$$

(1) General formulation

Different from the previous section, ρ is a non-linear transformation here. According to the structure of the model and the assumption between the GPs for modeling the $\rho_q\left(f_{q-1}(\boldsymbol{x})\right)$ and $\delta_q(\boldsymbol{x})$, the right hand side of Eq. (2.25) can be combined as a new GP whose inputs is x and $f^*_{q-1}(\boldsymbol{x})$. $f^*_{q-1}(\boldsymbol{x})$ denotes the sample from the posterior of the GP modeling at the $q-1$ fidelity evaluated at $\boldsymbol{x}$. Under this circumstance, Eq. (2.25) can be written in the following form:

$$f_q(\boldsymbol{x}) = g_q\left(f^*_{q-1}(\boldsymbol{x}), \boldsymbol{x}\right) \tag{2.26}$$

Although this scheme can be used to construct our nonlinear autoregressive algorithm, the covariance function of g_q's structure maybe have a different form because our inputs are $f^*_{q-1}(\boldsymbol{x})$ and $\boldsymbol{x}$ now. To reflect the autoregressive nature of Eq. (2.25), a more structured prior for g_q had been introduced . We consider the covariance kernel in the following form:

$$k_{q_g} = k_{q_\rho}\left(\boldsymbol{x}, \boldsymbol{x}'; \theta_{q_\rho}\right) \cdot k_{q_f}\left(f_{*_{q-1}}(\boldsymbol{x}), f_{*_{q-1}}\left(\boldsymbol{x}'\right); \theta_{q_f}\right) + k_{q_\delta}\left(\boldsymbol{x}, \boldsymbol{x}'; \theta_{q_\delta}\right) \tag{2.27}$$

Where k_{q_g}, k_{q_f} and k_{q_δ} are the covariance function and $\theta_{q_\rho}, \theta_{q_f}, \theta_{q_\delta}$ denote their hyperparameters.

(2) Non-linear multi-fidelity prediction

The lowest fidelity ($q = 1$) level of non-linear multi-fidelity scheme is trained by using the first fidelity data $\{\boldsymbol{x}_1, \boldsymbol{y}_1\}$. In this case, the posterior distribution follows a Gaussian distribution. The posterior's mean and variance can be got from Eq. (2.4) and (2.5), respectively. However, the following fidelities ($q \geq 2$) whose posterior distribution does not follow Gaussian distribution anymore, because the prediction can only be computed by using new the test input point $\left(\boldsymbol{x}_*, f_{*_{q-1}}\left(\boldsymbol{x}_*\right)\right)$. At this point, $f_{*_{q-1}}\left(\boldsymbol{x}_*\right)$ no longer follows Gaussian distribution. Note that it follows Gaussian distribution when $q = 2$. When $q \geq 2$, we need to obtain the prediction by giving new inputs $\boldsymbol{x}_*$. Then, the following form of the posterior distribution can be obtained:

$$
\begin{aligned}
p(f_{*_q}(\boldsymbol{x}_*)) &:= p(f_q(\boldsymbol{x}_*, f_{*_{q-1}}(\boldsymbol{x}_*)|f_{*_{q-1}}, \boldsymbol{x}_*, \boldsymbol{y}_q, \boldsymbol{x}_q) \\
&= \int p(f_q(\boldsymbol{x}_*, f_{*_{q-1}}(\boldsymbol{x}_*))|\boldsymbol{x}_*, \boldsymbol{y}_q, \boldsymbol{x}_q) p(f_{*_{q-1}}(\boldsymbol{x}_*)) d\boldsymbol{x}_*
\end{aligned} \tag{2.28}
$$

At this point, the dependence of all the hyper-parameters can be ignored, whereas $p\left(f_{*_{q-1}}\left(\boldsymbol{x}_*\right)\right)$ is the Gaussian posterior distribution at the fidelity level $(q-1)$. All posteriors $p\left(f_{*_q}\left(\boldsymbol{x}_*\right)\right), q \geq 2$'s predictive mean and variance can be calculated by using Monte Carlo integration of Eq. (2.28).

(3) Workflow of nonlinear multi-fidelity method

Now the workflow of this nonlinear autoregressive method will be introduced. Given the multi-fidelity input and output pairs $\left\{\boldsymbol{x}_q, \boldsymbol{y}_q\right\}$ which sorted in ascending fidelity's level $q = 1, \ldots, s$, and our datasets are nested and noiseless, then we perform the following steps:

Step 1: The lowest fidelity data $\{\boldsymbol{x}_1, \boldsymbol{y}_1\}$ and the kernel function $k_1\left(\boldsymbol{x}_1, \boldsymbol{x}_1'; \theta_1\right)$ can be employed to train the GP regression model Eq. (2.1) by maximizing the log-likelihood Eq. (2.2).

Step 2: For other fidelity levels from $q = 2, \ldots, s$, We use the data $\left\{\left(\boldsymbol{x}_q, f_{*_{q-1}}\left(\boldsymbol{x}_q\right)\right), \boldsymbol{y}_q\right\}$ and new kernel function Eq. (2.27) to train the new GP regression model Eq. (2.1) by maximizing the log-likelihood Eq. (2.2). The dimension of the GP model is $(d+1)$. In order to ensure the result converges to a local optimal, the gradient descend L-BFGS algorithm can be applied.

Step 3: $\left\{\left(\boldsymbol{x}_s, f_{*_{s-1}}\left(\boldsymbol{x}_s\right)\right), \boldsymbol{y}_s\right\}$ at s fidelity level can be used to train our last GP model, the new test point $\boldsymbol{x}^*$ can be applied to calculate the posterior predictive mean and variance by using the Monte Carlo integration of Eq. (2.28). This procedure needs to sample the posteriors at different fidelity levels $p\left(f_{*_q}\left(\boldsymbol{x}_*\right)\right), q = 1, \ldots, s$, and the output of the lower fidelity can be used as the input of the next-higher fidelity.

2.4 Bayesian Active Learning with GPs

In most cases, the size of low-fidelity observation is large and the number of high-fidelity data is limited or small. We usually want to add more data points to augment the original data size to make more accurate inferences, however, it is very hard for us to get high-fidelity data.

With the help of Bayesian active learning method, additional samples are obtained at the location where the maximum variance value is located. After obtaining the maximum variance value at samples $\boldsymbol{x}^*$, the new samples $\{\boldsymbol{x}^*, \boldsymbol{y}_L^*\}, \{\boldsymbol{x}^*, \boldsymbol{y}_H^*\}$ can be obtained. Finally, these new samples can be added to the original data.

2.4.1 Workflow of Bayesian active learning

To further improve the prediction accuracy, the Bayesian active learning has been employed, which can guide us where to evaluate a function next using the maximum variance criteria. In that case, the algorithm will be introduced step by step:

(a) Firstly, assuming our training data set consisting of N input-output observations:

$$\mathcal{D}_N = \left(\boldsymbol{x}^{1:N}, \boldsymbol{y}^{1:N}\right). \tag{2.29}$$

(b) For $N, N+1, \ldots,$ we start do the following: the current dataset $\mathcal{D}_N$ can be applied to quantify our state of knowledge about $f(\boldsymbol{x})$. For example, Gaussian process regression or any other Bayesian regression method can be used to obtain the predictive distribution:

$$f(\cdot)|\mathcal{D}_N \sim p(f(\cdot)|\mathcal{D}_N). \tag{2.30}$$

Then we pick the most informative sample to evaluate next by maximizing an acquisition function $a_N(\boldsymbol{x})$ which depends on our current state of knowledge. The acquisition function quantifies how much value or how much information is in eval-

uating $\boldsymbol{x}$. Assuming that it is a non-negative function. So, to pick the next sample, the problem can be denoted in the following form:

$$\boldsymbol{x}^{N+1} = \arg\max a_N(\boldsymbol{x}). \tag{2.31}$$

The next sample with the maximum value or the maximum information is picked. If the maximum value of the acquisition function is smaller than a threshold, then stop searching at this sample. Otherwise, we evaluate the function at the selected $\boldsymbol{x}^{N+1}$ to obtain:

$$\boldsymbol{y}^{N+1} = f(\boldsymbol{x}^{N+1}). \tag{2.32}$$

This process will let us obtain a dataset to minimize predictive variance, $(\boldsymbol{x}^{N+1}, \boldsymbol{y}^{N+1})$. the new observation can be got:

$$\mathcal{D}_{N+1} = \left((\boldsymbol{x}^{1:N}, \boldsymbol{x}^{N+1}), (\boldsymbol{y}^{1:N}, \boldsymbol{y}^{N+1})\right). \tag{2.33}$$

Bayes' rule can be applied to update our state of knowledge:

$$f(\cdot)|\mathcal{D}_{N+1} \sim p(f(\cdot)|\mathcal{D}_{N+1}) \propto p(\boldsymbol{y}^{N+1}|\boldsymbol{x}^{N+1}, f(\cdot))p(f(\cdot)|\mathcal{D}_N). \tag{2.34}$$

(c) At this point, we report our current state of knowledge about the maximum variance of the function. In a word, the index of where the maximum variance value is located can be found.

$$i^* = \arg\max_{1 \le i \le N} var(y^i), \tag{2.35}$$

And the maximum variance is $var(y^{i^*})$ and the location of the maximum is $\boldsymbol{x}^{i^*}$.

Note that currently the active learning is performed without dimension reduction, because we want to ensure the dimension reduction matrix $\boldsymbol{W}$ to be invertible.

2.4.2 Acquisition function

The acquisition function which used in Bayesian active learning will be introduced. There are several different acquisition functions, such as maximum upper interval, probability of improvement and expected improvement. In this paper, we focused on the maximum upper interval. It is defined to be:

$$a_N(\boldsymbol{x}) = \mu_N(\boldsymbol{x}) + \psi\sigma_N(\boldsymbol{x}), \tag{2.36}$$

Note that $\mu_N(\boldsymbol{x})$ is the predictive mean and $\sigma_N(\boldsymbol{x})$ is variance, $\psi \geq 0$. The parameter ψ controls how much emphasis you put on exploitation and exploration. The choice $\psi = 0$ is full-on exploitation. You are just looking at the predictive mean. The greater ψ is, the more emphasis you put on the predictive standard deviation, i.e., the more you try to explore. A large ψ value is used for finding where the maximum variance is located.

2.5 Summary of Multi-fidelity model with dimension reduction method

In the proposed method, we only consider two fidelities. Given our low- and high-fidelity training datasets $\{\boldsymbol{x}, \boldsymbol{y}_L\}, \{\boldsymbol{x}, \boldsymbol{y}_H\}$. A summary of the multi-fidelity with dimension reduction method is shown below:

Step 1: The original input low- and high-fidelity data is denoted as $\{\boldsymbol{x}, \boldsymbol{y}_L\}, \{\boldsymbol{x}, \boldsymbol{y}_H\}$, respectively. The low-fidelity data can be employed to calculate BIC score to determine the active subspace dimension d. Based on d, dimension reduction matrix $\boldsymbol{W}$ can be obtained based on the low-fidelity data.

Step 2: Secondly, according to the high-fidelity data, find where the largest variance is located, $\boldsymbol{x}^*$, by using Bayesian active learning method. We combine the new sample $\{\boldsymbol{x}^*, \boldsymbol{y}_L^*\}, \{\boldsymbol{x}^*, \boldsymbol{y}_H^*\}$ with our original data $\{\boldsymbol{x}, \boldsymbol{y}_L\}, \{\boldsymbol{x}, \boldsymbol{y}_H\}$. The new combined point can be denoted as $\left\{\boldsymbol{x}_{new}, \boldsymbol{y}_{L_{new}}\right\}, \left\{\boldsymbol{x}_{new}, \boldsymbol{y}_{H_{new}}\right\}$. Then we perform the dimension reduction on the Bayesian-active-learning-enriched new data, it can be written as $\left\{\boldsymbol{x}_{DR}, \boldsymbol{y}_{L_{new}}\right\}, \left\{\boldsymbol{x}_{DR}, \boldsymbol{y}_{H_{new}}\right\}$. Here $\boldsymbol{x}_{DR} = \boldsymbol{W}^T\boldsymbol{x}_{new}$.

Step 3: Apply the nonlinear multi-fidelity method to do the prediction, our input data is $\left\{\boldsymbol{x}_{DR}, \boldsymbol{y}_{L_{new}}\right\}, \left\{\boldsymbol{x}_{DR}, \boldsymbol{y}_{H_{new}}\right\}$. We use the low fidelity data $\left\{\boldsymbol{x}_{DR}, \boldsymbol{y}_{L_{new}}\right\}$ and kernel function $k_L\left(\boldsymbol{x}_{DR}, \boldsymbol{x}'_{DR}; \theta_L\right)$ to train GP regression model by maximizing the log-likelihood function.

Step 4: The data $\left\{\left(\boldsymbol{x}_{DR}, f_{*_L}\left(\boldsymbol{x}_{DR}\right)\right), \boldsymbol{y}_{H_{new}}\right\}$ and new kernel function Eq. (2.27) are employed to train our new GP regression model. The L-BFGS algorithm is applied to ensure the result converges to a local optimal.

Step 5: Finally, the new test sample is employed to obtain the posterior predictive mean μ_{pre} and variance σ^2_{pre} by calculating Eq. (2.28) according to Monte Carlo integration method. The new predicted output is μ_{pre}.

3. RESULT

In the previous section, the workflow of new proposed method had been introduced. In this section, some numerical examples will be presented to show the performance and accuracy of the proposed model. Linear and nonlinear autoregressive multi-fidelity model denote as AR1 and NARGP, respectively. We first calculate the BIC score to find the best active subspace dimension and use the gradient-free active subspace to get the dimension reduction matrix based on low-fidelity data. Secondly, the Bayesian active learning method is applied to add new samples according to the high-fidelity data into our original data. Then the data after dimension reduction or projected data is employed as the inputs of the NARGP model. To compare the performance of the NARGP to the AR1 method, we also use the data before dimension reduction or original data to train the NARGP and AR1 model. In this paper, the cases with two fidelities had been discussed. In order to check the performance of our model, some test samples for model prediction had been generated. Where DR stands for dimension reduction. The NARGP result (after DR) is our model result.

3.1 Validation example 1

This function is a four-dimensional test function:

$$f_H(\boldsymbol{x}) = \exp\left(x_1 + x_2 + x_3 + x_4\right) \tag{3.1}$$

$$f_L(\boldsymbol{x}) = x_4 f_H(\boldsymbol{x}) \tag{3.2}$$

Low- and high-fidelity training data $\{\boldsymbol{x}, \boldsymbol{y}_L\}, \{\boldsymbol{x}, \boldsymbol{y}_H\}$ can be obtained from $\boldsymbol{y}_L = f_L(\boldsymbol{x})$ and $\boldsymbol{y}_H = f_H(\boldsymbol{x})$, respectively. Our input variable is $\boldsymbol{x} = (x_1, ..., x_i)(i = 4)$, x_i follows the uniform distribution, we can draw training samples $\boldsymbol{x}$ from the interval

$[0,1]^4$. Here we take $N_L = 60$, $N_H = 20$, $N_{test} = 10$, where N_L and N_H denote the training data size of the $\{\boldsymbol{x}, \boldsymbol{y}_L\}$ and $\{\boldsymbol{x}, \boldsymbol{y}_H\}$, N_{test} is the number of test points. Note that Example 1 is a four-dimensional problem.

In order to identify the active subspace dimension, we first use the low-fidelity data as the observation data to get the BIC score of different input dimensions, as shown in Fig. 3.1.

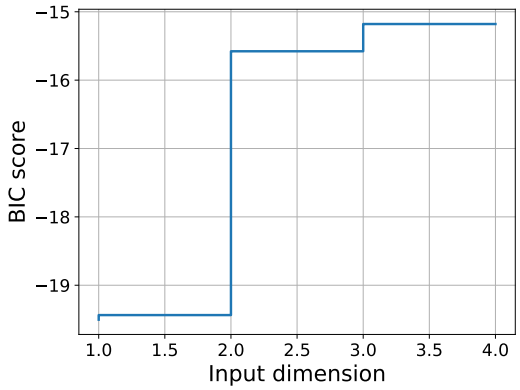


Figure 3.1. Example 1 - BIC score vs the input dimensions.

From Figure 3.1, the sharp increasing slope is at $d = 2$. Hence, the active subspace dimension is chosen as 2. In other words, original dimension can be decreased from $D = 4$ to $d = 2$. In that case, the dimension reduction matrix $\boldsymbol{W}$ is a 4×2 matrix.

$$\boldsymbol{W} = \begin{pmatrix} 0.28 & 0.50 \\ 0.28 & 0.50 \\ 0.28 & 0.50 \\ 0.87 & -0.49 \end{pmatrix} \tag{3.3}$$

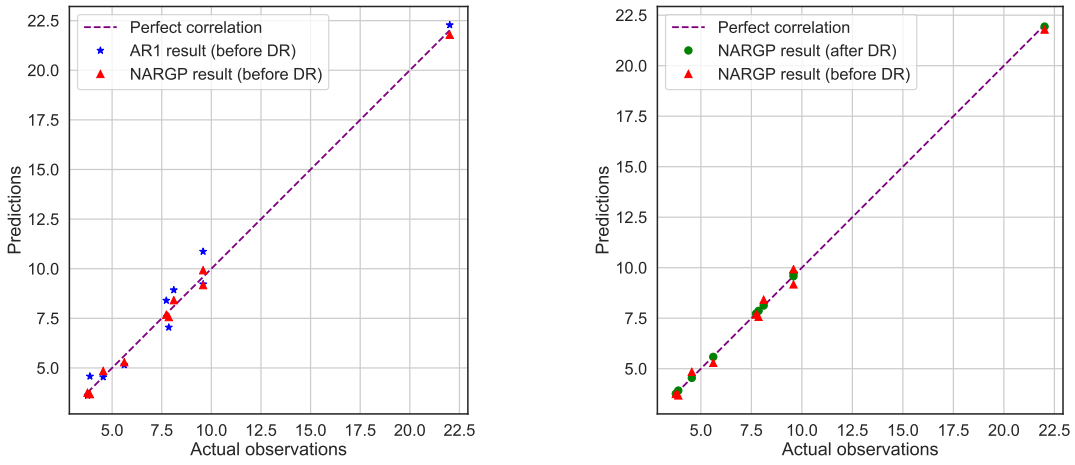
Bayesian active learning method is employed to add 10 more samples. The training data size becomes $N_L = 70$, $N_H = 30$. The model can be denoted as:

$$f_H(\boldsymbol{x}) = h_1(\boldsymbol{x}_{DR}) \tag{3.4}$$

Where h_1 is the mapping between data after dimension reduction and true observation. And $\boldsymbol{x}_{DR} = (\boldsymbol{x}_{DR1}, \boldsymbol{x}_{DR2})$, $\boldsymbol{x}_{DR1}$ is $\boldsymbol{x}$ times the first column of $\boldsymbol{W}$, $\boldsymbol{x}_{DR2}$ is $\boldsymbol{x}$ times the second column of $\boldsymbol{W}$.

$$\begin{aligned} \boldsymbol{x}_{DR1} &= 0.28x_1 + 0.28x_2 + 0.28x_3 + 0.87x_4 \\ \boldsymbol{x}_{DR2} &= 0.5x_1 + 0.5x_2 + 0.5x_3 - 0.49x_4 \end{aligned} \tag{3.5}$$

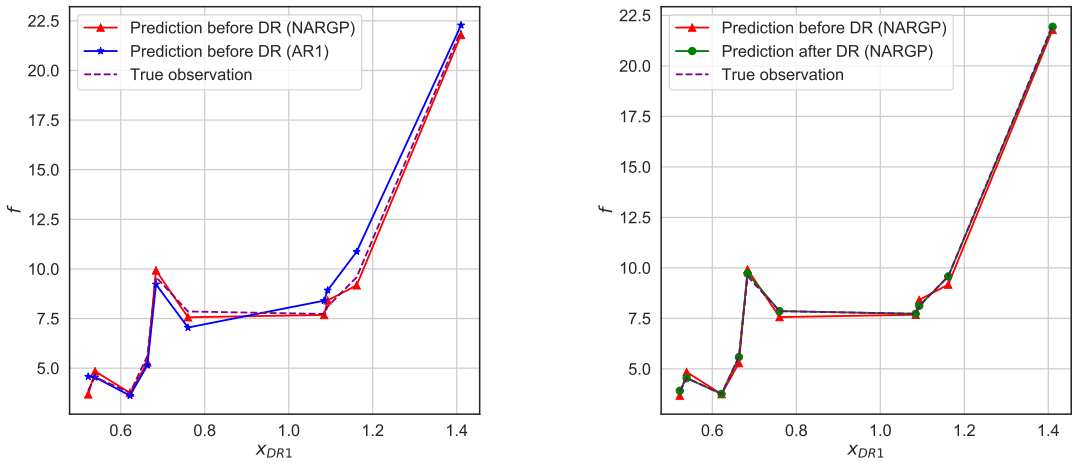
Finally, we use the data after dimension reduction ($d = 2$) as the inputs to run the proposed model. The data without dimension reduction ($D = 4$) is also employed as inputs to run the AR1 and NARGP methods for comparison. Numerical results are shown in Figs. 3.2, 3.3, 3.4 and 3.5.



(a) (b)

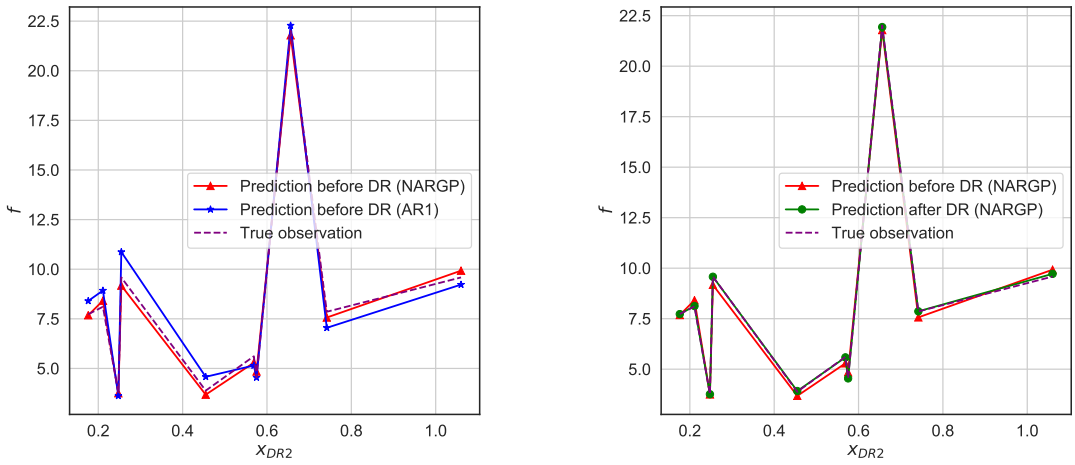
Figure 3.2. (a) Example 1 - Correlation before DR (AR1 and NARGP), (b) Correlation after DR (NARGP).

Figure 3.2 represents the correlation between the prediction and the true observation. Blue stars are the numerical results of AR1 method based on the original data ($D = 4$), the red triangles represent the results of NARGP method based on the original data ($D = 4$). Note that the training data size is $N_L = 60$, $N_H = 20$. Green dots are the proposed model results based on the data after dimension reduction ($d = 2$)



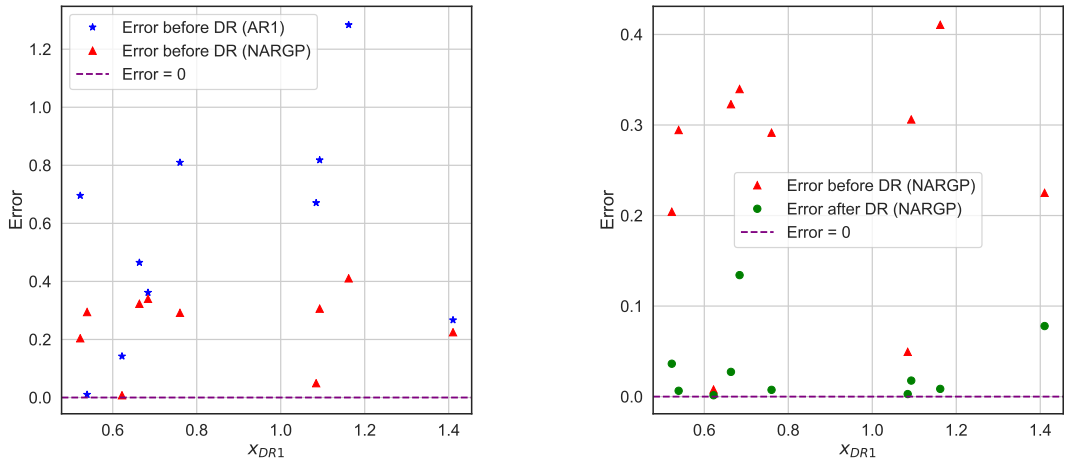
(a) (b)

Figure 3.3. Example 1 - (a) f vs x_{DR1} (AR1 and NARGP), (b) f vs x_{DR1} (NARGP).



(a) (b)

Figure 3.4. Example 1 - (a) f vs x_{DR2} (AR1 and NARGP), (b) f vs x_{DR2} (NARGP).



(a) (b)

Figure 3.5. Example 1 - (a) Error vs x_{DR1} (AR1 and NARGP), (b) Error vs x_{DR1} (NARGP).

and the training data size is $N_L = 70$, $N_H = 30$. Purple dash lines are the perfect correlation between the prediction and the true observation.

Figures 3.3 and 3.4 provide the prediction of f vs x_{DR1} and x_{DR2}, respectively. The blue line presents the numerical results of AR1 method based on the original data. The red line shows the numerical results of NARGP method based on the original data. Note that the training data size is $N_L = 60$, $N_H = 20$. The green line is the proposed model results based on the data after dimension reduction and the training data size is $N_L = 70$, $N_H = 30$. The purple dash line is the true observation vs x_{DR1} and x_{DR2}, respectively.

Figure 3.5 shows the error vs x_{DR1}. The error is defined as the absolute value of the prediction minus the true observation. The blue stars represent the numerical results of AR1 method based on the original data. The red triangles present the numerical results of NARGP method based on the original data. Note that the training data size is $N_L = 60$, $N_H = 20$. The green dots show the proposed model result based on the data after dimension reduction ($d = 2$) and the training data size is $N_L = 70$, $N_H = 30$.

As demonstrated in Figure 3.2 (a), the red triangles are closer to the perfect correlation line than the blue stars. As shown in Figures 3.3 (a) and 3.4 (a), the red line is nearly on the track of the true observation line. In addition, in Figure 3.5 (a), the red triangles are closer to the error = 0 line. Hence, the NARGP method performs better than the AR1 method based on the original data. As shown in Figure 3.2 (b), the green dots are closer to the perfect correlation line than the red triangles. As shown in Figures 3.3 (b) and 3.4 (b), the green line nearly matches with the true observation line. In addition, in Figure 3.5 (b), the green dots is closer to the error = 0 line. Hence, we can conclude that the proposed model results based on the data after dimension reduction is better than the NARGP method based on the original data.

3.2 Validation example 2

This function is a five-dimensional test function:

$$f_H(\boldsymbol{x}) = \exp\left(1.2(x_1 + x_2 + x_3)\right) \tag{3.6}$$

$$f_L(\boldsymbol{x}) = \sin\left(x_4 + x_5\right) f_H(\boldsymbol{x}) \tag{3.7}$$

Low- and high-fidelity training data $\{\boldsymbol{x}, \boldsymbol{y}_L\}, \{\boldsymbol{x}, \boldsymbol{y}_H\}$ can be acquired from $\boldsymbol{y}_L = f_L(\boldsymbol{x})$ and $\boldsymbol{y}_H = f_H(\boldsymbol{x})$, respectively. Our input variables $\boldsymbol{x} = (x_1, ..., x_i)(i = 5)$, x_i follows the uniform distribution. The training data points $\boldsymbol{x}$ are drawn from the interval $[0, 1]^5$. Here we take $N_L = 100$, $N_H = 30$, $N_{test} = 100$, where N_L and N_H denote the training data size of the $\{\boldsymbol{x}, \boldsymbol{y}_L\}$ and $\{\boldsymbol{x}, \boldsymbol{y}_H\}$, N_{test} is the number of test samples. Example 2 is a five-dimensional problem.

In order to identify the active subspace dimension, we first employ the low-fidelity data as the observation data to obtain the BIC score of different input dimensions as shown in Fig. 3.6.

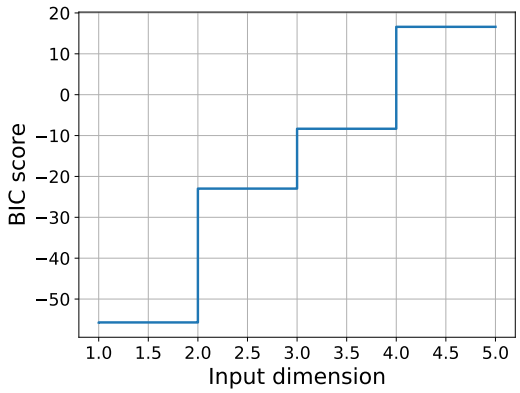


Figure 3.6. Example 2 - BIC score vs the input dimensions.

From Figure 3.6, the sharp increasing slope is at $d = 2$. The active subspace dimension is chosen as 2. In other words, the original dimension can be decreased

from $D = 5$ to $d = 2$. In that case, the dimension reduction matrix $\boldsymbol{W}$ is a 5×2 matrix.

$$\boldsymbol{W} = \begin{pmatrix} 0.17 & 0.55 \\ 0.17 & 0.55 \\ 0.17 & 0.55 \\ 0.68 & -0.21 \\ 0.68 & -0.21 \end{pmatrix} \tag{3.8}$$

Bayesian active learning method is employed to add 20 more samples. The training data size becomes $N_L = 120$, $N_H = 50$. The proposed model can be denoted as:

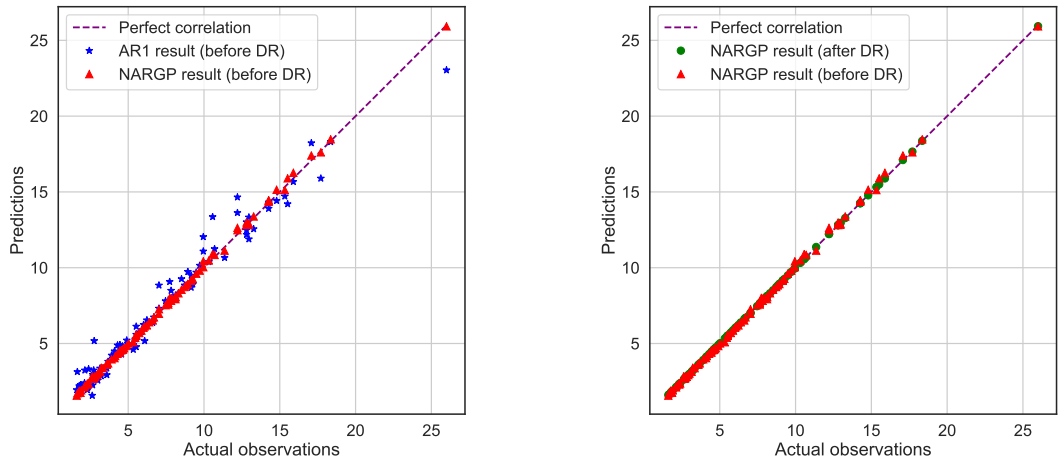
$$f_H(\boldsymbol{x}) = h_2(\boldsymbol{x}_{DR}) \tag{3.9}$$

Where h_2 is the mapping between the data after dimension reduction and the true observation. And $\boldsymbol{x}_{DR} = (\boldsymbol{x}_{DR1}, \boldsymbol{x}_{DR2})$, $\boldsymbol{x}_{DR1}$ is $\boldsymbol{x}$ times the first column of $\boldsymbol{W}$, $\boldsymbol{x}_{DR2}$ is $\boldsymbol{x}$ times the second column of $\boldsymbol{W}$.

$$\begin{aligned} \boldsymbol{x}_{DR1} &= 0.17x_1 + 0.17x_2 + 0.17x_3 + 0.68x_4 + 0.68x_5 \\ \boldsymbol{x}_{DR2} &= 0.55x_1 + 0.55x_2 + 0.55x_3 - 0.21x_4 - 0.21x_5 \end{aligned} \tag{3.10}$$

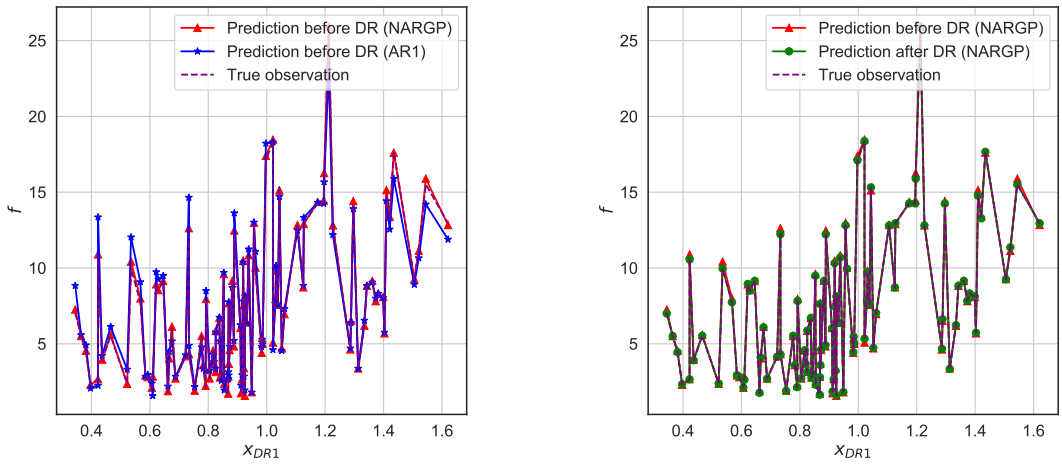
Finally, the data after dimension reduction ($d = 2$) is employed as the inputs to run the proposed model. The data without dimension reduction ($D = 5$) is employed as the inputs to run the AR1 and NARGP methods for prediction. The numerical results are shown in Fig. 3.7.

Figure 3.7 represents the correlation between the prediction and the true observation. The blue stars present the numerical results of the AR1 method based on the original data ($D = 5$). The red triangles show the numerical results of the NARGP method based on the original data ($D = 5$). Note that the training data size is $N_L = 100$, $N_H = 30$. The green dots present the proposed model results based on the data after dimension reduction ($d = 2$) and the training data size is $N_L = 120$,



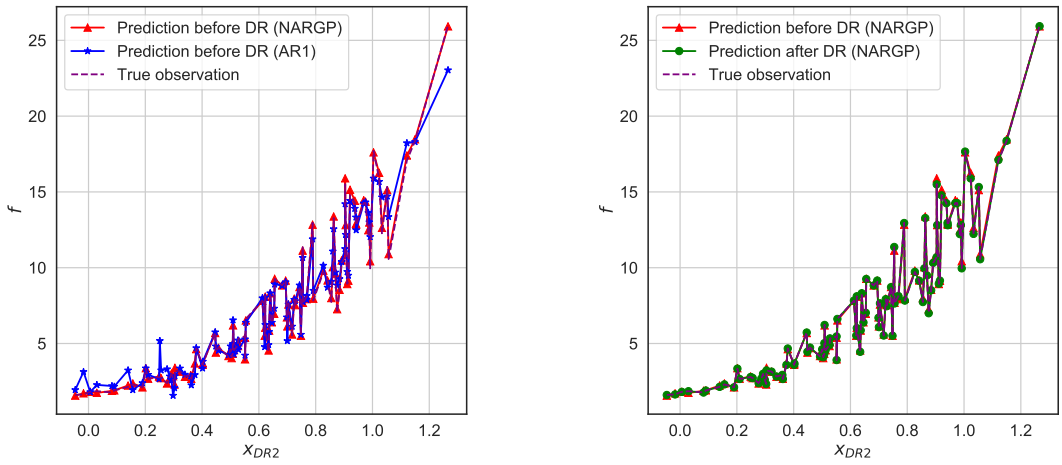
(a) (b)

Figure 3.7. Example 2 - (a) Correlation before DR (AR1 and NARGP), (b) Correlation after DR (NARGP).



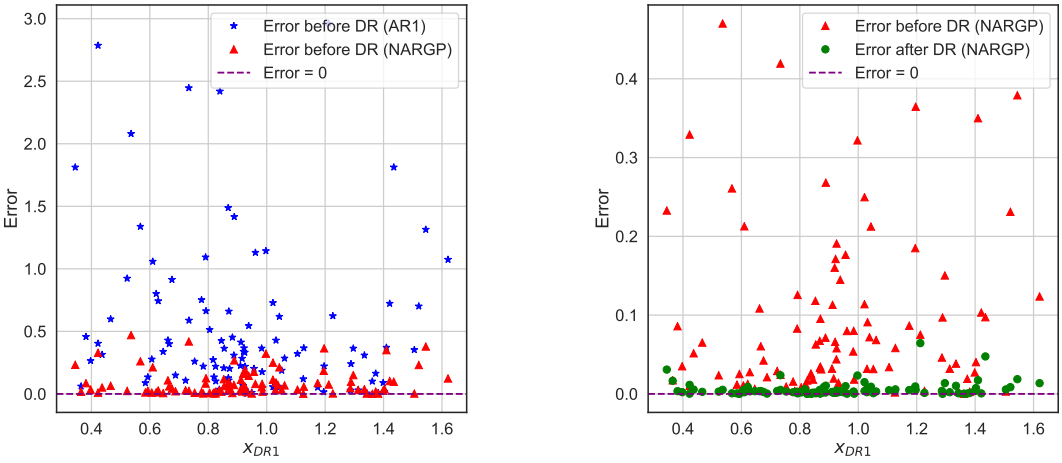
(a) (b)

Figure 3.8. Example 2 - (a) f vs x_{DR1} (AR1 and NARGP), (b) f vs x_{DR1} (NARGP).



(a) (b)

Figure 3.9. Example 2 - (a) f vs x_{DR2} (AR1 and NARGP), (b) f vs x_{DR2} (NARGP).



(a) (b)

Figure 3.10. Example 2 - (a) Error vs x_{DR1} (AR1 and NARGP), (b) Error vs x_{DR1} (NARGP).

$N_H = 50$. The purple dash line is the perfect correlation between the prediction and the true observation.

Figures 3.8 and 3.9 present the prediction f vs x_{DR1} and x_{DR2}, respectively. The blue line shows the numerical results of the AR1 method based on the original data. The red line is the numerical results of the NARGP method based on the original data. Note that the training data size is $N_L = 100$, $N_H = 30$. The green line represents the proposed model results based on the data after dimension reduction and the training data size is $N_L = 120$, $N_H = 50$. The purple dash line shows the true observation vs x_{DR1} and x_{DR2}, respectively.

Figure 3.10 shows the error vs x_{DR1}. The error is defined as the absolute value of the prediction minus the true observation. The blue stars show the numerical results of the AR1 method based on the original data. The red triangles are the numerical results of the NARGP method based on the original data. Note that the training data size is $N_L = 100$, $N_H = 30$. The green dots are the proposed model results based on the data after dimension reduction ($d = 2$) and the training data size is $N_L = 120$, $N_H = 50$.

As shown in Figure 3.7 (a), the red triangle is closer to the perfect correlation line than the blue stars. As shown in Figures 3.8 (a) and 3.9 (a), the red line is nearly on the track of the true observation line. In addition, in Figure 3.5 (a), the red triangles are closer to the error = 0 line. Hence, the NARGP method performs better than the AR1 method based on the original data.

From Figure 3.7 (b), it is hard to tell which results are better. As shown in Figures 3.8 (b) and 3.9 (b), it seems the red and the green lines match with the true observation line well. In addition, in Figure 3.10 (b), the green dots are closer to the error = 0 line. Hence, we can conclude that the proposed model results based on the data after dimension reduction is better than the NARGP method based on the original data.

3.3 Poisson equation

In this numerical example, ten-dimensional (10D) Poisson equation has been introduced and it's shown in Eq. (3.11). The source term $f(\boldsymbol{x})$ is assumed to follow the form of Eq. (3.12). The solution $u(\boldsymbol{x})$ of the Poisson equation follows the form of Eq. (3.13).

$$\sum_{d=1}^{10} \frac{\partial^2}{\partial x_d^2} u(\boldsymbol{x}) = f(\boldsymbol{x}) \tag{3.11}$$

$$\begin{cases} f_H(\boldsymbol{x}) = -32\pi^2 \sin\left(2\pi(x_1 + x_3)\right) \\ f_L(\boldsymbol{x}) = 0.8 f_H(\boldsymbol{x}) - 4\pi^2 \sin\left(2\pi x_2\right). \end{cases} \tag{3.12}$$

$$\begin{cases} u_H(\boldsymbol{x}) = 4 \sin\left(2\pi(x_1 + x_3)\right) \\ u_L(\boldsymbol{x}) = 0.8 u_H(\boldsymbol{x}) + \sin\left(2\pi x_2\right). \end{cases} \tag{3.13}$$

Low- and high-fidelity training data $\{\boldsymbol{x}, \boldsymbol{u}_L\}, \{\boldsymbol{x}, \boldsymbol{u}_H\}$ can be got from $u_L(\boldsymbol{x})$ and $u_H(\boldsymbol{x})$, respectively. The input variables are $\boldsymbol{x} = (x_1, ..., x_i)(i = 10)$. Here x_i follows the uniform distribution. The training data points $\boldsymbol{x}$ can be drawn from the interval $[0, 1]^{10}$. Here we take $N_L = 150$, $N_H = 20$, $N_{test} = 100$, where N_L and N_H denote the training data size of the $\{\boldsymbol{x}, \boldsymbol{u}_L\}$ and $\{\boldsymbol{x}, \boldsymbol{u}_H\}$, N_{test} is the number of test points.

In order to identify the active subspace dimension, we first use the low-fidelity data to get BIC score of different input dimensions, as shown in Fig. 3.11.

In Figure 3.11, the sharp slope is at $d = 2$. The active subspace dimension is chosen as 2. In other words, we can decrease the original dimension from $D = 10$ to $d = 2$. In that case, the dimension reduction matrix $\boldsymbol{W}$ is a 10×2 matrix.

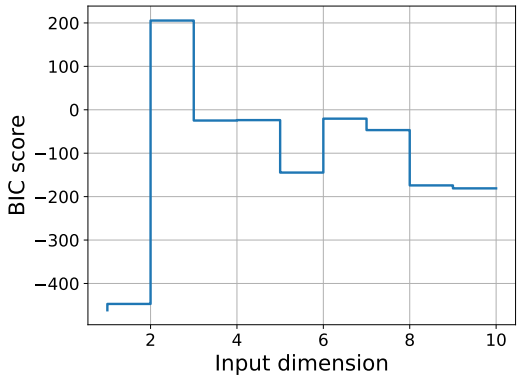


Figure 3.11. Poisson equation - BIC score vs the input dimensions.

$$\boldsymbol{W} = \begin{pmatrix} 0.71 & 0.02 \\ -0.02 & 1 \\ 0.71 & 0.02 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{3.14}$$

Bayesian active learning method is employed to add 30 more samples. The training data size becomes $N_L = 180$, $N_H = 50$. The proposed model can be denoted as:

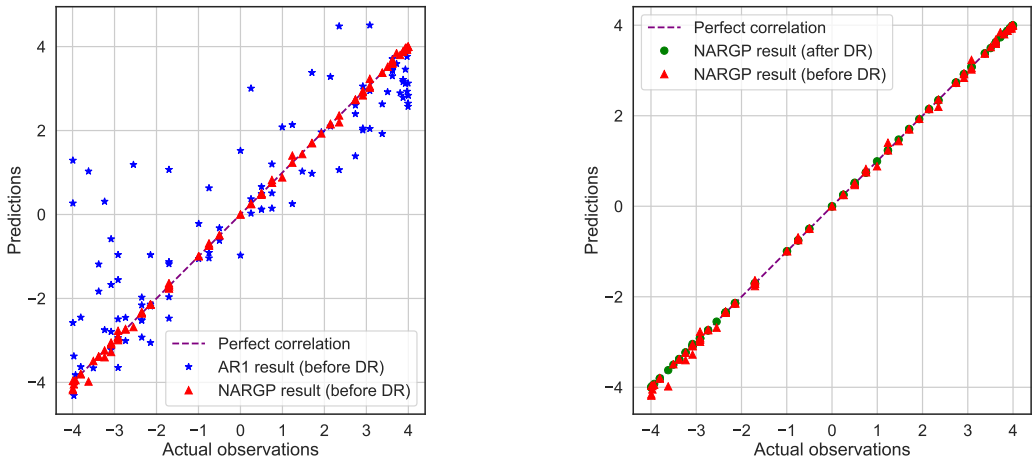
$$u_H(\boldsymbol{x}) = h_3(\boldsymbol{x}_{DR}) \tag{3.15}$$

Where h_3 is the mapping between the data after dimension reduction and the true observation. We denote $\boldsymbol{W} = (\boldsymbol{W}_1, \boldsymbol{W}_2)$, $\boldsymbol{W}_1$ is the first column of $\boldsymbol{W}$, $\boldsymbol{W}_2$ is the second column of $\boldsymbol{W}$. $\boldsymbol{x}_{DR} = (\boldsymbol{x}_{DR1}, \boldsymbol{x}_{DR2})$:

$$\begin{aligned} \boldsymbol{x}_{DR1} &= \boldsymbol{W}_1^T \boldsymbol{x} \\ \boldsymbol{x}_{DR2} &= \boldsymbol{W}_2^T \boldsymbol{x} \end{aligned} \tag{3.16}$$

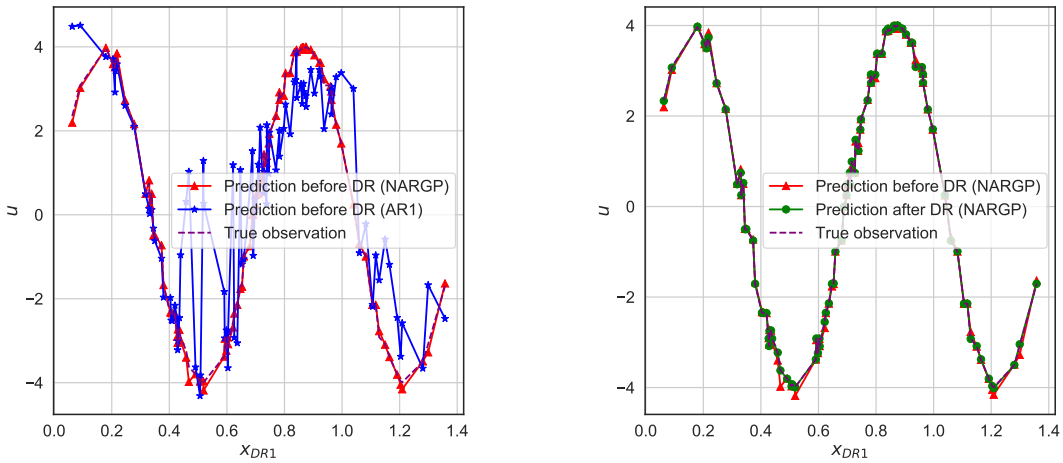
Finally, the data after dimension reduction ($d = 2$) is employed as the inputs to run the proposed model. The data without dimension reduction ($D = 10$) is also employed as the inputs to run the AR1 and NARGP models for prediction. Results are shown in Fig. 3.12.

Figure 3.12 represents the correlation between the prediction and the true observation. The blue stars represent the numerical results of the AR1 method based on the original data ($D = 10$). The red triangles show the numerical results of the NARGP method based on the original data ($D = 10$). Note that the training



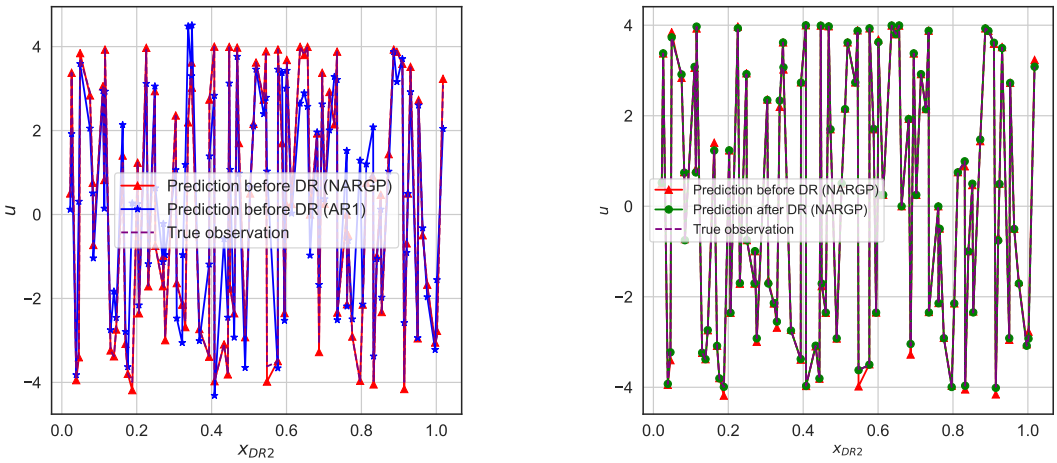
(a) (b)

Figure 3.12. Poisson equation - (a) Correlation before DR (AR1 and NARGP), (b) Correlation after DR (NARGP).



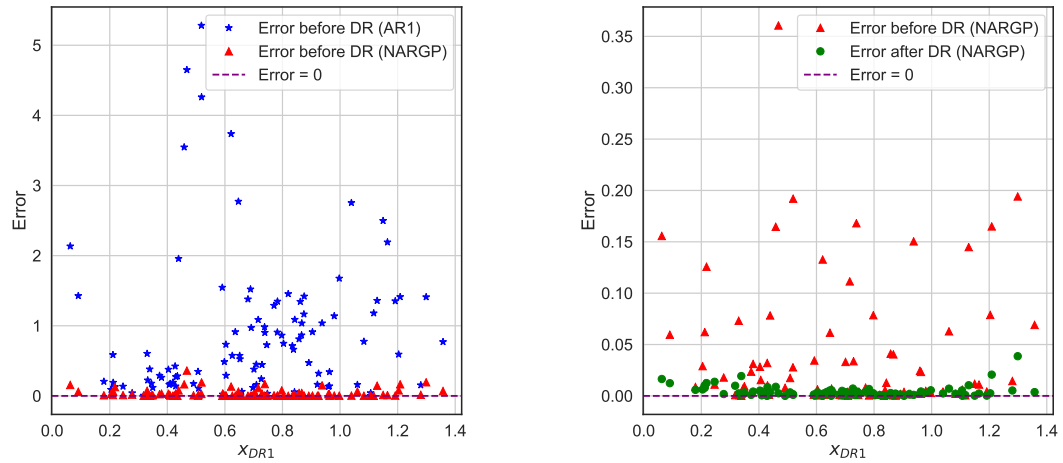
(a) (b)

Figure 3.13. Poisson equation - (a) u vs x_{DR1} (AR1 and NARGP), (b) u vs x_{DR1} (NARGP).



(a) (b)

Figure 3.14. Poisson equation - (a) u vs x_{DR2} (AR1 and NARGP), (b) u vs x_{DR2} (NARGP).



(a) (b)

Figure 3.15. Poisson equation - (a) Error vs x_{DR1} (AR1 and NARGP), (b) Error vs x_{DR1} (NARGP).

data size is $N_L = 150$, $N_H = 20$. The green dots show the proposed model results based on the data after dimension reduction ($d = 2$) and the training data size is $N_L = 180$, $N_H = 50$. The purple dash line represents the perfect correlation between the prediction and the true observation.

Figures 3.13 and 3.14 present the prediction u vs x_{DR1} and x_{DR2}, respectively. The blue line presents the numerical results of the AR1 method based on the original data. The red line shows the numerical results of the NARGP method based on the original data. Note that the training data size is $N_L = 150$, $N_H = 20$. The green line is the proposed model results based on the data after dimension reduction and the training data size is $N_L = 180$, $N_H = 50$. The purple dash line presents the true observation vs x_{DR1} and x_{DR2}, respectively.

Figure 3.15 shows the error vs x_{DR1}. Error is defined as the absolute value of prediction minus true observation. The blue stars show the numerical results of the AR1 method based on the original data. The red triangles present the numerical results of the NARGP method based on the original data. Note that the training data size is $N_L = 150$, $N_H = 20$. The green dots show the proposed model results based on the data after dimension reduction ($d = 2$) and the training data size is $N_L = 180$, $N_H = 50$.

In Figure 3.12 (a), the red triangles are closer to the perfect correlation line than the blue stars. The blue stars are totally off from the perfect correlation line. As shown in Figures 3.13 (a) and 3.14 (a), the red line is nearly on the track of the true observation line. In addition, in Figure 3.15 (a), the red triangles are closer to the error = 0 line. Hence, the NARGP method performs better than the AR1 method based on the original data.

In Figure 3.12 (b), it is hard to tell which results are better. As shown in Figures 3.13 (b) and 3.14 (b), it seems the red and the green lines match with the true observation line well. In addition, in Figure 3.15 (b), the green dots are closer to the error = 0 line. Hence, we can conclude that the proposed model results based on the

data after dimension reduction are better than the NARGP model results based on the original data.

3.4 KdV equation

In this example, the Korteweg-de Vries (KdV) equation [41–46] is studied:

$$\frac{\partial u}{\partial t}(x,t;\boldsymbol{\xi}) - 6u(x,t;\boldsymbol{\xi})\frac{\partial u}{\partial x}(x,t;\boldsymbol{\xi}) + \frac{\partial^3 u}{\partial x^3}(x,t;\boldsymbol{\xi}) = f(t;\boldsymbol{\xi}), x \in (-\infty,+\infty) \tag{3.17}$$

$$u(x,0;\boldsymbol{\xi}) = -2\,\mathrm{sech}^2(x) \tag{3.18}$$

Defining

$$W(t;\boldsymbol{\xi}) = \int_0^t f(y;\boldsymbol{\xi})dy \tag{3.19}$$

The analytical solution of the KdV equation is:

$$u(x,t;\boldsymbol{\xi}) = W(t;\boldsymbol{\xi}) - 2\,\mathrm{sech}^2\left(x - 4t + 6\int_0^t W(z;\boldsymbol{\xi})dz\right) \tag{3.20}$$

We define $f(t;\boldsymbol{\xi})$ as a Gaussian random field and KL expansion (KLE) [47] can be employed to represent it below:

$$f(t;\boldsymbol{\xi}) = \sigma\sum_{i=1}^{d}\sqrt{\lambda_i}\phi_i(t)\boldsymbol{\xi}_i \tag{3.21}$$

Where σ is a constant, $\boldsymbol{\xi} = \xi_i, i = 1,...,d_{max}$ and $\{\lambda_i, \phi_i(t)\}_{i=1}^{d_{max}}$ are eigenpairs of the exponential covariance kernel $C(x,x')$. In this problem, we set $l_c = 0.25$.

$$C(x,x') = \exp\left(\frac{|x-x'|}{l_c}\right) \tag{3.22}$$

In this case, the exact one-soliton solution is:

$$u(x,t;\boldsymbol{\xi}) = \sigma\sum_{i=1}^{d_{max}}\sqrt{\lambda_i}\boldsymbol{\xi}_i\int_0^t\phi_i(y)dy - 2\,\mathrm{sech}^2\left(x - 4t + 6\sigma\sum_{i=1}^{d_{max}}\sqrt{\lambda_i}\boldsymbol{\xi}_i\int_0^t\int_0^z\phi_i(y)dydz\right) \tag{3.23}$$

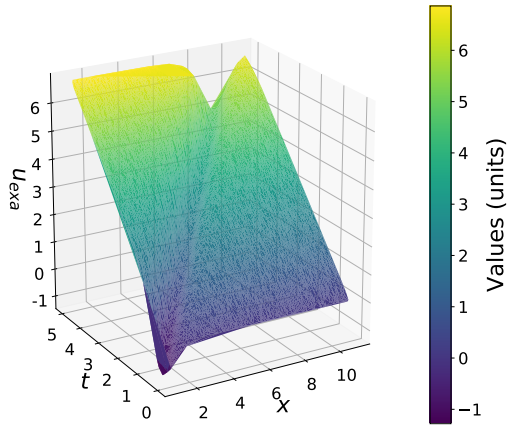


Figure 3.16. 3D Contour of the KdV equation.

Since the expression for the ϕ_i is known, it helps us to calculate the integrals in the equation above more accurately.

Denoting

$$\begin{aligned} A_i &= \sqrt{\lambda_i} \int_0^t \phi_i(y) dy \\ B_i &= \sqrt{\lambda_i} \int_0^t \int_0^z \phi_i(y) dy dz \end{aligned} \tag{3.24}$$

Here the quantities of interest is $u(x, t; \boldsymbol{\xi})$ at $x = 6$, $t = 1$, $\sigma = 1$. The analytical solution is:

$$u(\boldsymbol{\xi}) = u(x, t; \boldsymbol{\xi})|_{x=6,t=1} = \sum_{i=1}^{d_{max}} A_i \boldsymbol{\xi}_i - 2\,\mathrm{sech}^2 \left(2 + 6 \sum_{i=1}^{d_{max}} B_i \boldsymbol{\xi}_i \right) \tag{3.25}$$

The 3D contour of Eq. (3.23) with fixed ξ is shown in Figure 3.16. The three-axes are x, t and u, respectively. Here $x \in [1, 11]$, $t \in [0, 5]$.

According to Eq. (3.25), the low-fidelity data $(\boldsymbol{\xi}, \boldsymbol{u}_L)$ is obtained by setting $d_{max} = 2$, and high-fidelity data $(\boldsymbol{\xi}, \boldsymbol{u}_H)$ is obtained by setting $d_{max} = 10$. The input variables are $\boldsymbol{\xi} = (\xi_1, ..., \xi_i)(i = 10)$. ξ_i follows the uniform distribution. The training samples $\boldsymbol{\xi}$ are drawn from the interval $[0, 1]^{10}$. Here we take $N_L = 10$, $N_H = 3$ and $N_{test} = 50$ where N_L and N_H denote the number of training low- and high-fidelity data separately, N_{test} means the number of testing data.

In order to identify the active subspace dimension, the low-fidelity data is employed to obtain the BIC score of different input dimensions, as shown in Fig. 3.17.

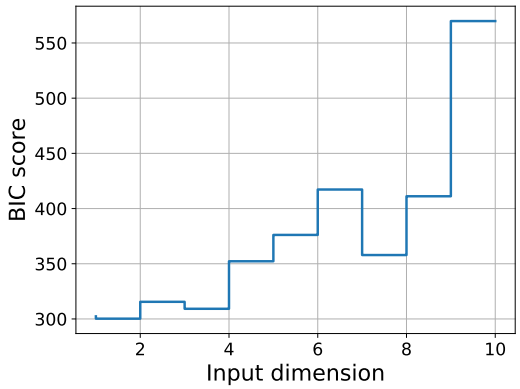


Figure 3.17. KdV equation - BIC score vs the input dimensions.

In Figure 3.17, $d = 2$ is chosen according to the BIC score criterion. The active subspace dimension is chosen as 2. In other words, the original dimension can be reduced from $D = 10$ to $d = 2$. In that case, the dimension reduction matrix $\boldsymbol{W}$ is a 10×2 matrix.

Bayesian active learning method is employed to add 30 more samples. The training data size becomes $N_L = 40$, $N_H = 33$. The proposed model can be denoted as:

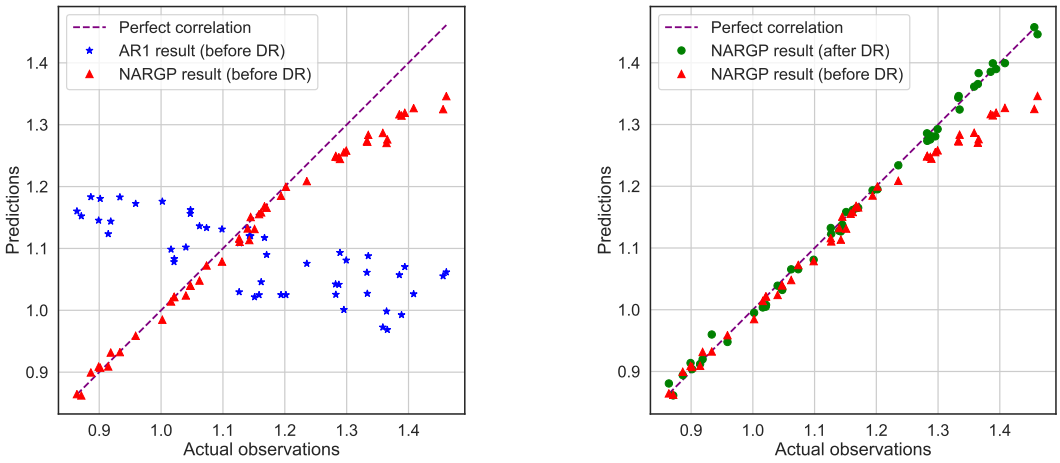
$$u_H(\boldsymbol{\xi}) = h_4(\boldsymbol{\xi}_{DR}) \tag{3.26}$$

Where h_4 is the mapping between the data after dimension reduction and the true observation. And $\boldsymbol{W} = (\boldsymbol{W}_1, \boldsymbol{W}_2)$, $\boldsymbol{W}_1$ is the first column of $\boldsymbol{W}$, $\boldsymbol{W}_2$ is the second column of $\boldsymbol{W}$. $\boldsymbol{\xi}_{DR} = (\boldsymbol{\xi}_{DR1}, \boldsymbol{\xi}_{DR2})$:

$$\begin{aligned} \boldsymbol{\xi}_{DR1} &= \boldsymbol{W}_1^T \boldsymbol{\xi} \\ \boldsymbol{\xi}_{DR2} &= \boldsymbol{W}_2^T \boldsymbol{\xi} \end{aligned} \tag{3.27}$$

Finally, the data after dimension reduction ($d = 2$) is employed as inputs to run the proposed model. The data without dimension reduction ($D = 10$) is also

employed as inputs to run the AR1 and NARGP models for prediction. Results are shown in Fig. 3.18.

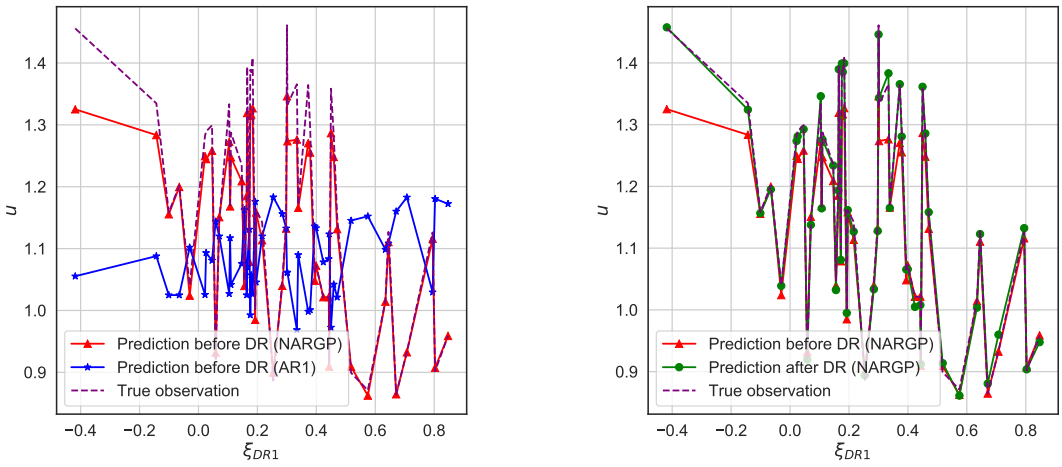


(a) (b)

Figure 3.18. KdV equation - (a) Correlation before DR (AR1 and NARGP), (b) Correlation after DR (NARGP).

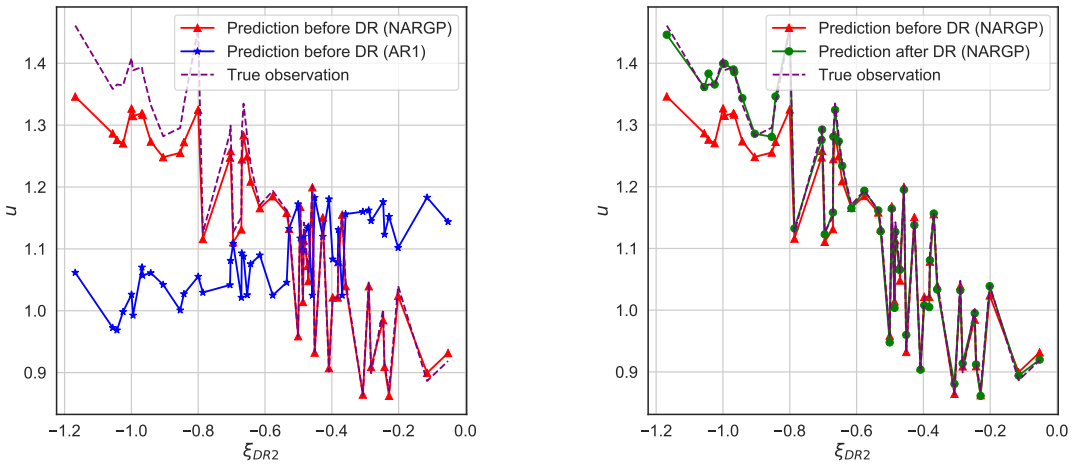
Figure 3.18 represents the correlation between the prediction and the true observation. The blue stars present the numerical results of the AR1 method based on the original data ($D = 10$). The red triangles show the numerical results of the NARGP method based on the original data ($D = 10$). Note that the training data size is $N_L = 10$, $N_H = 3$. The green dots are the proposed model results based on the data after dimension reduction ($d = 2$) and the training data size is $N_L = 40$, $N_H = 33$. The purple dash line means the perfect correlation between prediction and true observations.

Figures 3.19 and 3.20 provide the prediction u vs ξ_{DR1} and ξ_{DR2}, respectively. The blue line presents the result of the AR1 method based on original data. The red line shows the numerical results of the NARGP method based on the original data. Note that the training data size is $N_L = 10$, $N_H = 3$. The green line presents the proposed model result based on data after dimension reduction and the training data size is



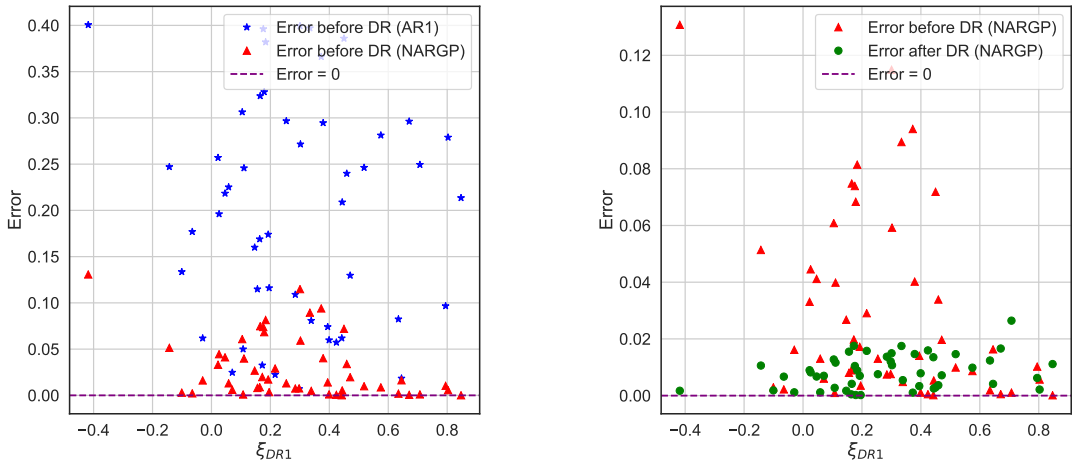
(a) (b)

Figure 3.19. KdV equation - (a) u vs ξ_{DR1} (AR1 and NARGP), (b) u vs ξ_{DR1} (NARGP).



(a) (b)

Figure 3.20. KdV equation - (a) u vs ξ_{DR2} (AR1 and NARGP), (b) u vs ξ_{DR2} (NARGP).



(a) (b)

Figure 3.21. KdV equation - (a) Error vs ξ_{DR1} (AR1 and NARGP), (b) Error vs ξ_{DR1} (NARGP).

$N_L = 40$, $N_H = 33$. The purple dash line shows the true observation vs x_{DR1} and x_{DR2}, respectively.

Figure 3.21 shows the error vs ξ_{DR1}. Error is defined the absolute value of prediction minus true observations. The blue stars present the numerical results of the AR1 method based on original data, the red triangle is the numerical results of the NARGP method based on the original data, Here we need to note that the training data size is $N_L = 10$, $N_H = 3$. The green dots represent the proposed model result based on data after dimension reduction ($d = 2$) and the training data size is $N_L = 40$, $N_H = 33$.

As shown in Figure 3.18 (a), the red triangles are closer to the perfect correlation line than the blue stars. The blue stars are totally off from the perfect correlation line. As shown in Figures 3.19 (a) and 3.20 (a), the red line is nearly on the track of true observation line. In addition, in Figure 3.21 (a), the red triangles are closer to the error = 0 line. Hence, the NARGP method performs better than the AR1 method based on the original data.

As shown in Figure 3.18 (b), the green dots distributed on the perfect correlation. The red triangles are off from the perfect situation. As shown in Figures 3.19 (b) and 3.20 (b), the green line matches with the true observation better than the red line. In addition, in Figure 3.21 (b), the green dots are closer to the error = 0 line. Hence, we can conclude that the proposed model results based on the data after dimension reduction is better than the NARGP model results based on the original data.

3.5 Elliptic equation

In this example, the elliptic differential equation with a random high-order coefficient is considered:

$$\begin{aligned} &-\tfrac{d}{dx}\left(a(x;\boldsymbol{\xi})\tfrac{du(x;\boldsymbol{\xi})}{dx}\right) = 1, \quad x \in (0,1) \\ &u(0) = u(1) = 0 \end{aligned} \tag{3.28}$$

Where $a(x;\boldsymbol{\xi})$ is a log-normal random field based on KL expansion:

$$a(x;\boldsymbol{\xi}) = a_0(x) + \exp\left(\sigma \sum_{i=1}^{d_{max}} \sqrt{\lambda_i}\phi_i(x)\xi_i\right) \tag{3.29}$$

where ξ_i are i.i.d. standard Gaussian random variables, $\{\lambda_i, \phi_i(x)\}_{i=1}^{d_{max}}$ are the largest eigenvalues and the corresponding eigenfunctions of the exponential covariance kernel in the form of Eq. (3.22).

In the KL expansion, λ_i denotes the eigenvalue of the covariance kernel $C(x, x')$. With this setting, a and u only depend on x and the solution of the deterministic elliptic equation can be written in the following form:

$$u(x) = u(0) + \int_0^x \frac{a(0)u(0)' - y}{a(y)}\mathrm{d}y \tag{3.30}$$

And our boundary condition is $u(0) = u(1) = 0$, $a(0)u(0)'$ can be computed:

$$a(0)u(0)' = \left(\int_0^1 \frac{y}{a(y)}\mathrm{d}y\right) / \left(\int_0^1 \frac{1}{a(y)}\mathrm{d}y\right) \tag{3.31}$$

In this example, we set $a_0(x) = 0.1$, $\sigma = 0.2$, $l_c = 0.2$, $x = 0.35$. The equation based on the equations (3.29) and (3.30) can rewritten in the following form:

$$u(\boldsymbol{\xi}) = \int_0^{0.35} \frac{a(0)u(0)' - \boldsymbol{\xi}}{a(\boldsymbol{\xi})}\mathrm{d}\boldsymbol{\xi} \tag{3.32}$$

According to Eq. (3.29), low-fidelity data $(\boldsymbol{\xi}, \boldsymbol{u}_L)$ and the high-fidelity data $(\boldsymbol{\xi}, \boldsymbol{u}_H)$ can be obtained by setting $a_L(x;\boldsymbol{\xi})$ and $a_H(x;\boldsymbol{\xi})$ in the following form:

$$a_L(x;\boldsymbol{\xi}) = a_0(x) + \exp\left(0.2\sum_{i=1}^{8} \sqrt{\lambda_i}\phi_i(x)\xi_i\right) + \exp\left(0.2\sum_{i=9}^{10} \sqrt{\lambda_i}\phi_i(x)\xi_i\right) \tag{3.33}$$

$$a_H(x;\boldsymbol{\xi}) = a_0(x) + \exp\left(0.2\sum_{i=1}^{8} \sqrt{\lambda_i}\phi_i(x)\xi_i\right) \tag{3.34}$$

Our input variables are $\boldsymbol{\xi} = (\xi_1, ..., \xi_i)(i = 10)$. ξ_i follows the uniform distribution. Training samples $\boldsymbol{\xi}$ can be drawn from the interval $[0, 1]^{10}$. Here we take $N_L = 15$, $N_H = 3$ and $N_{test} = 50$ where N_L and N_H denote the number of training low- and high-fidelity data separately, N_{test} means the number of testing data.

In order to identify the active subspace dimension, the low-fidelity data is employed to get the BIC score of different input dimensions, as shown in Fig. 3.22.

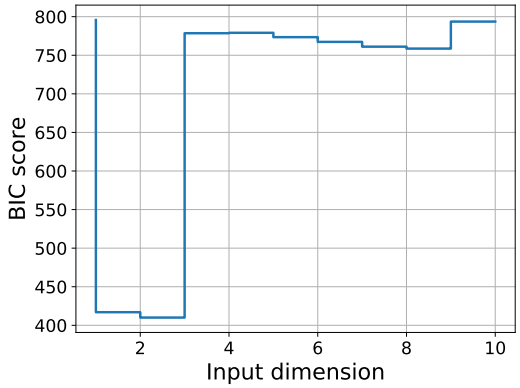


Figure 3.22. Elliptic equation - BIC score vs the input dimensions.

In Figure 3.22, the sharp slope is at $d = 3$. The active subspace dimension is 3. In other words, the original dimensions can be decrease from $D = 10$ to $d = 3$. In that case, the dimension reduction matrix $\boldsymbol{W}$ is a 10×3 matrix.

Bayesian active learning method is employed to add 20 more samples. The proposed training data size becomes $N_L = 35$, $N_H = 23$. The proposed model can be denoted as:

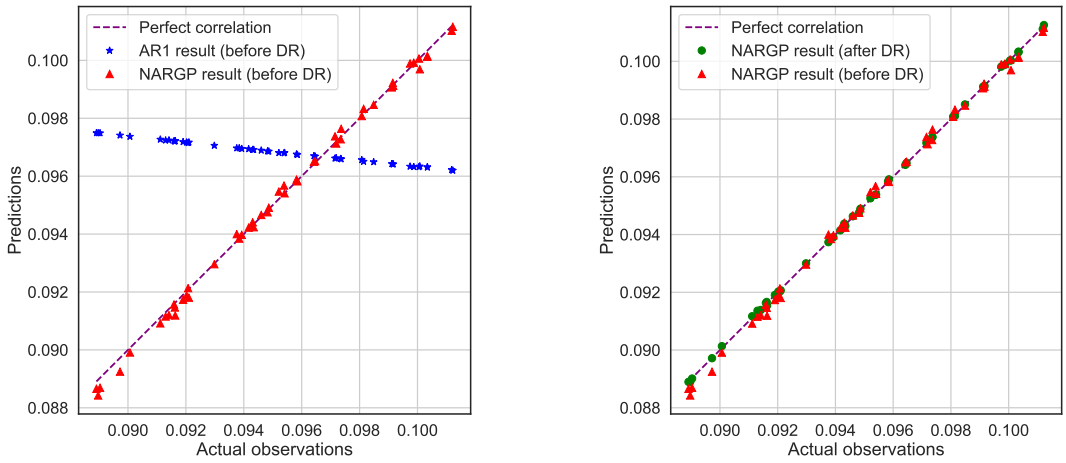
$$u_H(\boldsymbol{\xi}) = h_5(\boldsymbol{\xi}_{DR}) \tag{3.35}$$

Where h_5 is the mapping between data after dimension reduction and the true observation. And $\boldsymbol{W} = (\boldsymbol{W}_1, \boldsymbol{W}_2)$, $\boldsymbol{W}_1$ is the first column of $\boldsymbol{W}$, $\boldsymbol{W}_2$ is the second column of $\boldsymbol{W}$, $\boldsymbol{W}_3$ is the third column of $\boldsymbol{W}$. $\boldsymbol{\xi}_{DR} = (\boldsymbol{\xi}_{DR1}, \boldsymbol{\xi}_{DR2}, \boldsymbol{\xi}_{DR3})$:

$$\begin{aligned} \boldsymbol{\xi}_{DR1} &= \boldsymbol{W}_1^T \boldsymbol{\xi} \\ \boldsymbol{\xi}_{DR2} &= \boldsymbol{W}_2^T \boldsymbol{\xi} \\ \boldsymbol{\xi}_{DR3} &= \boldsymbol{W}_3^T \boldsymbol{\xi} \end{aligned} \tag{3.36}$$

Finally, the data after dimension reduction ($d = 3$) is employed as the inputs to run the proposed model. The data without dimension reduction ($D = 10$) is used

as inputs to run the AR1 and NARGP models for comparison. Results are shown in Fig. 3.23.

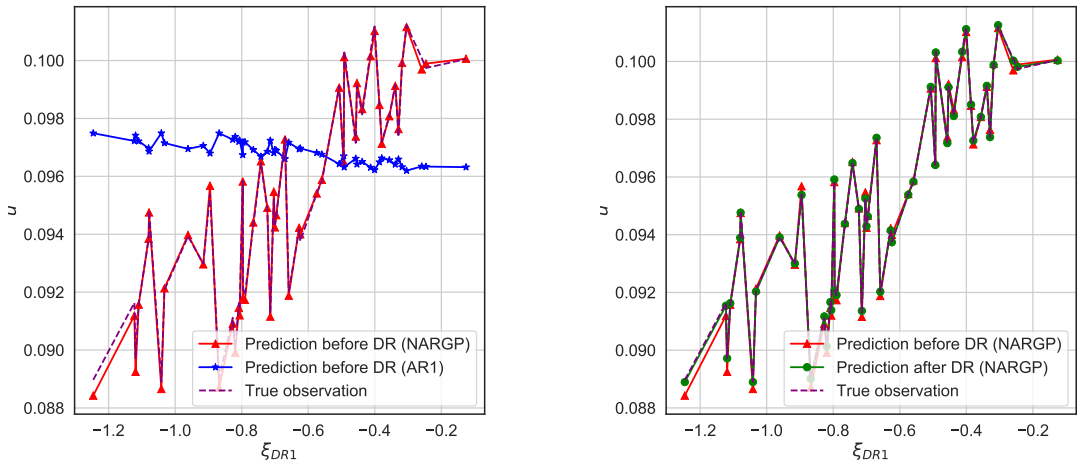


(a) (b)

Figure 3.23. Elliptic equation - (a) Correlation before DR (AR1 and NARGP), (b) Correlation after DR (NARGP).

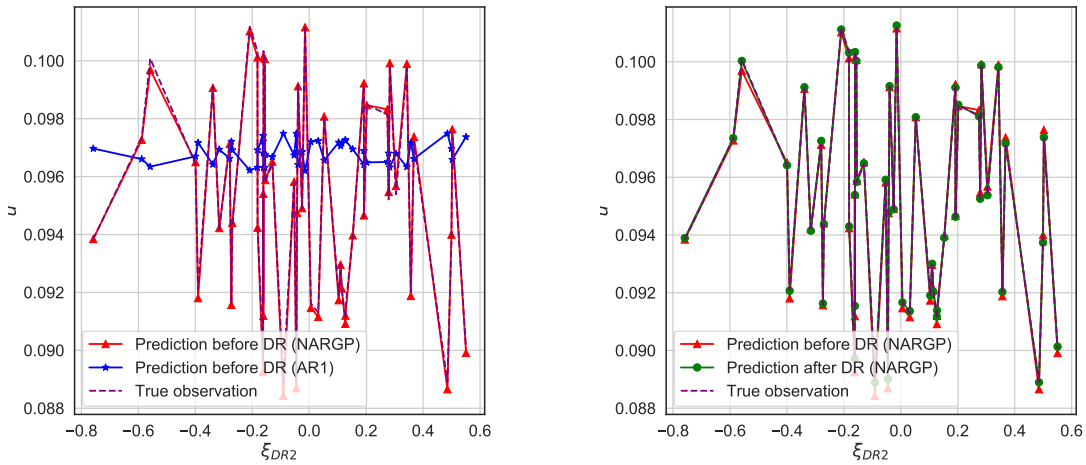
Figure 3.23 represents the correlation between the prediction and the true observation. The blue stars present the numerical results of the AR1 method based on the original data ($D = 10$), the red triangles show the numerical results of NARGP method based on the original data ($D = 10$). Note that the training data size is $N_L = 15$, $N_H = 3$. The green dots are the proposed model results based on data after dimension reduction ($d = 3$) and the training data size is $N_L = 35$, $N_H = 23$. The purple dash line represents the perfect correlation between prediction and true observations.

Figures 3.24, 3.25 and 3.26 provide the prediction u vs ξ_{DR1}, ξ_{DR2} and ξ_{DR3}, respectively. The blue line presents the numerical results of the AR1 method based on the original data. The red line shows the numerical results of the NARGP method based on the original data. Note that the training data size is $N_L = 15$, $N_H = 3$. The green line represents the proposed model results based on the data after dimension



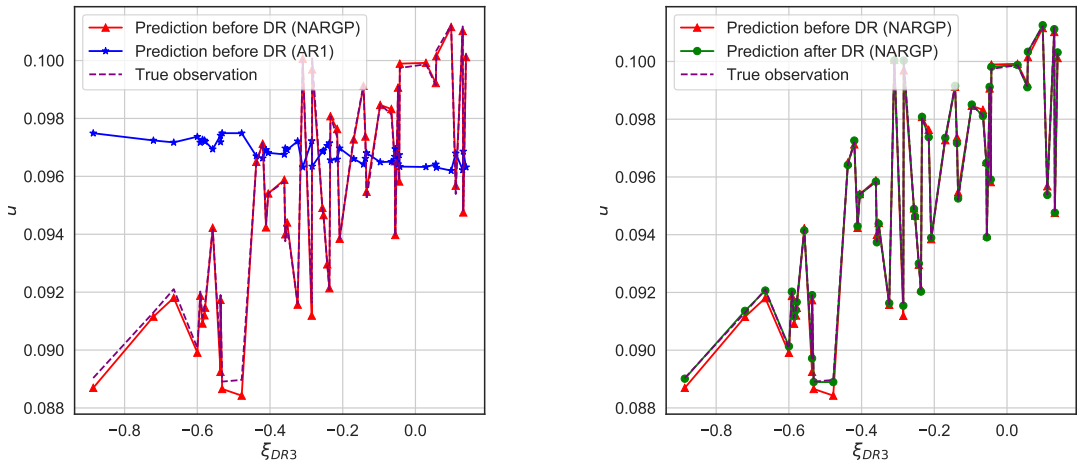
(a) (b)

Figure 3.24. Elliptic equation - (a) u vs ξ_{DR1} (AR1 and NARGP), (b) u vs ξ_{DR1} (NARGP).



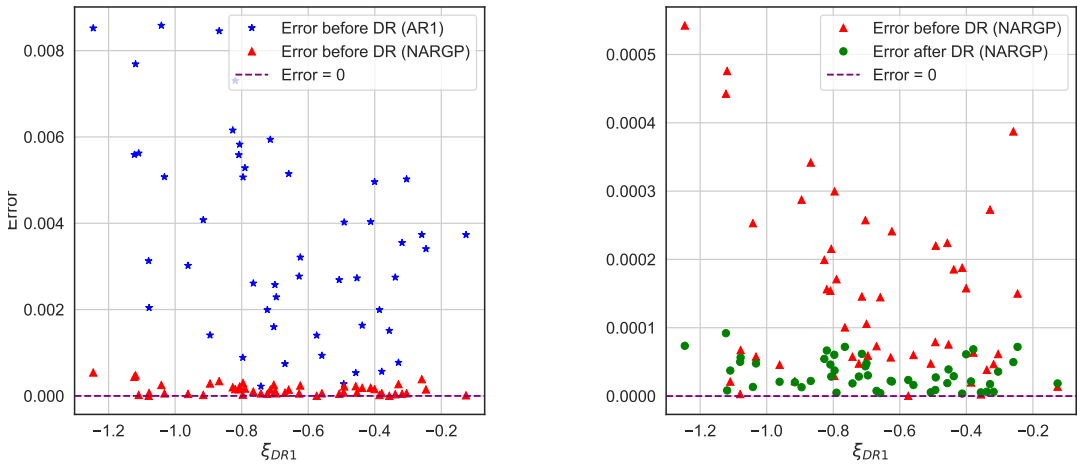
(a) (b)

Figure 3.25. Elliptic equation - (a) u vs ξ_{DR2} (AR1 and NARGP), (b) u vs ξ_{DR2} (NARGP).



(a) (b)

Figure 3.26. Elliptic equation - (a) u vs ξ_{DR3} (AR1 and NARGP), (b) u vs ξ_{DR3} (NARGP).



(a) (b)

Figure 3.27. Elliptic equation - (a) Error vs ξ_{DR1} (AR1 and NARGP), (b) Error vs ξ_{DR1} (NARGP).

reduction and the training data size is $N_L = 35$, $N_H = 23$. The purple dash line shows the true observation vs ξ_{DR1}, ξ_{DR2} and ξ_{DR3}, respectively.

Figure 3.27 shows the error vs ξ_{DR1}. Error is defined as the absolute value of prediction minus true observation. The blue star presents the numerical results of the AR1 method based on the original data. The red triangles show the numerical results of the NARGP method based on the original data. Note that the training data size is $N_L = 15$, $N_H = 3$. The green dots are the proposed model results based on the data after dimension reduction ($d = 3$) and the training data size is $N_L = 35$, $N_H = 23$.

From Figure 3.23 (a), the red triangles are closer to the perfect correlation line than the blue stars. The blue stars are totally off from the perfect correlation line. As shown in Figures 3.24 (a), 3.25 (a) and 3.26 (a), the red line presents nearly on the track of true observation line. In addition, in Figure 3.27 (a), the red triangles are closer to the error = 0 line. Hence, the NARGP method perform better than the AR1 method based on the original data.

From Figure 3.23 (b), it is hard to tell which results are better. As shown in Figures 3.24 (b), 3.25 (b) and 3.26 (b), the green line matches with the true observation better than the red line. In addition, in Figure 3.27 (b), the green dots are closer to the error = 0 line. Hence, we can draw the conclusion that the proposed model results based on the data after dimension reduction are better than the NARGP model results based on the original data.

3.6 Summary of mean square error

In addition, the mean square error (MSE) of the different situation had been summarized below, NARGP (after DR) is our model results. DR stands for dimension reduction:

It can be observed from Table 3.1 that the NARGP method can make better prediction than the AR1 method when the data before dimension reduction is inputs.

Table 3.1.
MSE of AR1 and NARGP model prediction with data before dimension reduction

MSE	AR1 (before DR)	NARGP (before DR)
Simulation 1	0.4346	0.0749
Simulation 2	0.6523	0.0185
Poisson equation	1.8189	0.0046
KdV equation	0.0539	0.0019
Elliptic equation	1.7861×10^{-5}	3.8738×10^{-8}

Table 3.2.
MSE of NARGP model prediction with data before and after dimension reduction

MSE	NARGP (before DR)	NARGP (after DR)
Simulation 1	0.0749	0.0027
Simulation 2	0.0185	0.0001
Poisson equation	0.0046	4.3551×10^{-5}
KdV equation	0.0019	0.0001
Elliptic equation	3.8738×10^{-8}	1.4233×10^{-9}

It is shown in Table 3.2 that the proposed model has better performance than the NARGP method which is based on data without dimension reduction. From all the results shown in this section, we draw conclusion that the proposed model can not only perform dimension reduction but also make accurate prediction.

4. CONCLUSION

In this work, a novel nonlinear multi-fidelity surrogate model is presented by integrating the advantage of the gradient-free active subspace method and the NARGP multi-fidelity scheme. The proposed model can not only reduce the input dimensions but also make accurate model prediction based on the data after dimension reduction. Several numerical examples are investigated to show that the proposed model can outperform the AR1 and NARGP multi-fidelity methods. In practice, the active subspace dimension value varies in different cases. From simulation results in this paper, it is a coincidence that most of the active subspace dimension is 2. Although the multi-fidelity model is widely used in the engineering applications, we have to acknowledge that the multi-fidelity method has not been fully explored and developed in the machine learning [48–50] field. We hope our work can inspire some research interest in this field.

REFERENCES

REFERENCES

[1] A. I. Forrester, A. Sóbester, and A. J. Keane, "Multi-fidelity optimization via surrogate modelling," *Proceedings of the royal society a: mathematical, physical and engineering sciences*, vol. 463, no. 2088, pp. 3251–3269, 2007.

[2] B. Peherstorfer, K. Willcox, and M. Gunzburger, "Survey of multifidelity methods in uncertainty propagation, inference, and optimization," *Siam Review*, vol. 60, no. 3, pp. 550–591, 2018.

[3] L. W. Ng and K. E. Willcox, "Multifidelity approaches for optimization under uncertainty," *International Journal for numerical methods in Engineering*, vol. 100, no. 10, pp. 746–772, 2014.

[4] M. G. Fernández-Godino, C. Park, N.-H. Kim, and R. T. Haftka, "Review of multi-fidelity models," *arXiv preprint arXiv:1609.07196*, 2016.

[5] C. Park, R. T. Haftka, and N. H. Kim, "Remarks on multi-fidelity surrogates," *Structural and Multidisciplinary Optimization*, vol. 55, no. 3, pp. 1029–1050, 2017.

[6] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[7] M. C. Kennedy and A. O'Hagan, "Predicting the output from a complex computer code when fast approximations are available," *Biometrika*, vol. 87, no. 1, pp. 1–13, 2000.

[8] I. Bilionis and N. Zabaras, "Multidimensional adaptive relevance vector machines for uncertainty quantification," *SIAM Journal on Scientific Computing*, vol. 34, no. 6, pp. B881–B908, 2012.

[9] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker, "A bayesian approach to the design and analysis of computer experiments," Oak Ridge National Lab., TN (USA), Tech. Rep., 1988.

[10] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments," *Statistical science*, pp. 409–423, 1989.

[11] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker, "Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments," *Journal of the American Statistical Association*, vol. 86, no. 416, pp. 953–963, 1991.

[12] I. Bilionis and N. Zabaras, "Multi-output local gaussian process regression: Applications to uncertainty quantification," *Journal of Computational Physics*, vol. 231, no. 17, pp. 5718–5746, 2012.

[13] B. A. Lockwood and M. Anitescu, "Gradient-enhanced universal kriging for uncertainty propagation," *Nuclear Science and Engineering*, vol. 170, no. 2, pp. 168–195, 2012.

[14] I. Bilionis, N. Zabaras, B. A. Konomi, and G. Lin, "Multi-output separable gaussian process: Towards an efficient, fully bayesian paradigm for uncertainty quantification," *Journal of Computational Physics*, vol. 241, pp. 212–239, 2013.

[15] I. Bilionis and N. Zabaras, "Solution of inverse problems with limited forward solver evaluations: a bayesian perspective," *Inverse Problems*, vol. 30, no. 1, p. 015004, 2013.

[16] P. Perdikaris, M. Raissi, A. Damianou, N. Lawrence, and G. E. Karniadakis, "Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 473, no. 2198, p. 20160751, 2017.

[17] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[18] X. Ma and N. Zabaras, "Kernel principal component analysis for stochastic input model generation," *Journal of Computational Physics*, vol. 230, no. 19, pp. 7311–7331, 2011.

[19] R. Tripathy, I. Bilionis, and M. Gonzalez, "Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation," *Journal of Computational Physics*, vol. 321, pp. 191–223, 2016.

[20] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint arXiv:1012.2599*, 2010.

[21] P. G. Constantine, "A quick-and-dirty check for a one-dimensional active subspace," *arXiv preprint arXiv:1402.3838*, 2014.

[22] P. Constantine, M. Emory, F. Palacios, N. Kseib, and G. Iaccarino, "Quantification of margins and uncertainties using an active subspace method for approximating bounds," in *11th International Conference on Structural Safety & Reliability*, 2013.

[23] P. Constantine and D. Gleich, "Computing active subspaces with monte carlo," *arXiv preprint arXiv:1408.0545*, 2014.

[24] P. G. Constantine, B. Zaharatos, and M. Campanelli, "Discovering an active subspace in a single-diode solar cell model," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 8, no. 5-6, pp. 264–273, 2015.

[25] P. G. Constantine, E. Dow, and Q. Wang, "Active subspace methods in theory and practice: applications to kriging surfaces," *SIAM Journal on Scientific Computing*, vol. 36, no. 4, pp. A1500–A1524, 2014.

[26] J. L. Jefferson, J. M. Gilbert, P. G. Constantine, and R. M. Maxwell, "Active subspaces for sensitivity analysis and dimension reduction of an integrated hydrologic model," *Computers & Geosciences*, vol. 83, pp. 127–138, 2015.

[27] E. Dow and Q. Wang, "Output based dimensionality reduction of geometric variability in compressor blades," in *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013, p. 420.

[28] T. W. Lukaczyk, P. Constantine, F. Palacios, and J. J. Alonso, "Active subspaces for shape optimization," in *10th AIAA multidisciplinary design optimization conference*, 2014, p. 1171.

[29] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[30] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," 1970.

[31] H. Haario, M. Laine, A. Mira, and E. Saksman, "Dram: efficient adaptive mcmc," *Statistics and computing*, vol. 16, no. 4, pp. 339–354, 2006.

[32] G. Golub and C. Van Loan, "Matrix computations," *Matrix*, vol. 1000, no. 13, p. 09, 1996.

[33] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on scientific computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[34] Z. Wen and W. Yin, "A feasible method for optimization with orthogonality constraints," *Mathematical Programming*, vol. 142, no. 1-2, pp. 397–434, 2013.

[35] L. Le Gratiet and J. Garnier, "Recursive co-kriging model for design of computer experiments with multiple levels of fidelity," *International Journal for Uncertainty Quantification*, vol. 4, no. 5, 2014.

[36] A. O'Hagan, "A markov property for covariance structures," *Statistics Research Report*, vol. 98, p. 13, 1998.

[37] P. Perdikaris and G. E. Karniadakis, "Model inversion via multi-fidelity bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond," *Journal of The Royal Society Interface*, vol. 13, no. 118, p. 20151107, 2016.

[38] X. Meng and G. E. Karniadakis, "A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems," *Journal of Computational Physics*, vol. 401, p. 109020, 2020.

[39] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Inferring solutions of differential equations using noisy multi-fidelity data," *Journal of Computational Physics*, vol. 335, pp. 736–746, 2017.

[40] K. Cutajar, M. Pullin, A. Damianou, N. Lawrence, and J. González, "Deep gaussian processes for multi-fidelity modeling," *arXiv preprint arXiv:1903.07320*, 2019.

[41] X. Yang, H. Lei, N. A. Baker, and G. Lin, "Enhancing sparsity of hermite polynomial expansions by iterative rotations," *Journal of Computational Physics*, vol. 307, pp. 94–109, 2016.

[42] G. Lin, L. Grinberg, and G. E. Karniadakis, "Numerical studies of the stochastic korteweg-de vries equation," *Journal of Computational Physics*, vol. 213, no. 2, pp. 676–703, 2006.

[43] G. Xu, G. Lin, and J. Liu, "Rare-event simulation for the stochastic korteweg–de vries equation," *SIAM/ASA Journal on Uncertainty Quantification*, vol. 2, no. 1, pp. 698–716, 2014.

[44] M. D. Kruskal, "The korteweg-de vries equation and related evolution equations," in *In: Nonlinear wave motion.(A75-14987 04-70) Providence, RI, American Mathematical Society, 1974, p. 61-83.*, 1974, pp. 61–83.

[45] R. Hirota and J. Satsuma, "Soliton solutions of a coupled korteweg-de vries equation," *Physics Letters A*, vol. 85, no. 8-9, pp. 407–408, 1981.

[46] R. Hirota, "Exact solution of the korteweg—de vries equation for multiple collisions of solitons," *Physical Review Letters*, vol. 27, no. 18, p. 1192, 1971.

[47] R. G. Ghanem and P. D. Spanos, *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.

[48] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, 2015.

[49] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[50] P. P. Raut and N. R. Borkar, "Machine learning algorithms: Trends, perspectives and prospects," *International Journal of Engineering Science*, vol. 4884, 2017.