Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 54-4 (2021) 56–61

Gaussian Process Modelling of the F-16 Ground Vibration Test Benchmark: Data Selection Case Study ⋆

Matija Perne * **Martin Stepančič** ** **Juš Kocijan** *,***

* *Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia*
** *Danfoss Trata, Ulica Jožeta Jame 16, 1210 Ljubljana – Šentvid, Slovenia*
*** *University of Nova Gorica, Vipavska 13, 5000 Nova Gorica, Slovenia*

Abstract: We observe the effects of training data sample selection in modelling of a physical system with Gaussian process nonlinear autoregressive models with exogenous input. Gaussian process modelling limits the number of training data points and we use a big nonlinear benchmark data set. The combination calls for training data sample selection. We compare a 'smart' method based on Euclidean distance between training data points with decimation. We use the training data samples obtained by both methods to train the models, test model predictions on a test data set, and calculate two figures of merit, $e_{\mathrm{RMS}t}$ and mean standardised log loss (MSLL). The model trained on the 'smartly' selected training data points is better in $e_{\mathrm{RMS}t}$ while the one with the decimated data is superior in MSLL. The direct conclusion is that the purpose of the model determines which training data sample selection method is better, as the relevant figure of merit depends on the model purpose. We notice that the predicted variance is more sensitive to the training data sample than the predicted mean. We warn that training data sample selection may have unexpected consequences.

Keywords: Gaussian processes, Data reduction, Benchmark examples, Nonlinear models, Probabilistic models, Modelling errors, Statistics

1. INTRODUCTION

The F-16 aircraft benchmark example data (Schoukens and Noël, 2015) was obtained by shaking an aircraft with two dummy payloads attached. The interfaces between the aircraft and the payloads caused nonlinear distortions. The provided signals are the input voltage to the shaker under the right wing, the force of the shaker, and acceleration at 3 points on the aircraft and the payload, all sampled at 400 Hz. Multiple experiments were done with various sizes and shapes of the input signal (Noël and Schoukens, 2017). In the Gaussian process (GP) modelling data selection case study, we use the shaker force and the acceleration on the right wing next to the nonlinear interface signals from the two highest excitation level multisine experiments with full frequency grid.

Gaussian process is a non-parametric kernel model (Rasmussen and Williams, 2006), the main comparative advantage of which is that it provides information on output uncertainty in addition to the predicted output value. The additional information comes at a computational cost, so only a modest number of training data points can be utilized (Kocijan, 2016) although significant progress is being made on increasing the number (Wang et al., 2019). The benchmark example offers an amount of data sufficient to overwhelm a typical GP model, so we are only able to use a part of the available data.

When only a sample of the available data can be used in training, there are different ways of choosing the sample. Apart from decimation – using every n-th training data point, where n is the decimation factor (Naghizadeh and Sacchi, 2010) – there are several other possibilities, for example based on convex hulls (Khosravani et al., 2016), K-means clustering (Tang et al., 2019), outlier pattern analysis and prediction (Lin et al., 2015), Markov geometric diffusion (Silva et al., 2016), etc. We test an algorithm based on Euclidean distances between the points in the training data set (Perne et al., 2019) on the benchmark example. It rejects data points that have close neighbours. It is easy to implement and modest enough in its use of computational resources that it can be used on the studied data set. We explore how the choice of the training data points influences the quality of the model. In particular, we set to find out whether a better model results from the training data sample chosen by decimation or by the advanced method.

The contribution of this paper is the study of the influence of data sample selection on the resulting model. Among the many data sample selection methods in existence, the one used is chosen because of the low effort it requires.

⋆ The authors acknowledge the financial support from the Slovenian Research Agency (project "Method for the forecasting of local radiological pollution of atmosphere using Gaussian process models", ID L2-8174, "Modelling the Dynamics of Short-Term Exposure to Radiation", ID L2-2615, and research core funding No. P2-0001).

10.1016/j.ifacol.2021.10.010

We present the data set, explain the regressor selection and the data sampling, and introduce the GP modelling and the figures of merit in Section 2. In Section 3, the model prediction results with the traditional and the innovative data sampling methods are compared. In part 4, we try to find meaning in the obtained results, and in 5, we conclude that a good data sample is a relative term and that data sample selection has to be done carefully because it gives one a significant freedom to influence the resulting model. The choice of the data sample selection method particularly strongly influences the predicted variance. Which data sample selection method is better depends on the figure of merit used to evaluate the resulting model.

2. METHODS

An overview of GP modelling is described in 2.1 and the NARX (nonlinear autoregressive model with exogenous input) structure in 2.2. The used data set is introduced in 2.3. The regressor and data sample selection follow in 2.4 and 2.5, respectively. Finally, the formulas used in evaluating the model results are given in 2.6.

We use the F-16 aircraft benchmark example data set as the dynamic system data to model. The system is nonlinear and the quantity of data is sufficient to call for data sample selection when training a GP.

2.1 Gaussian process modelling

GP is a generalisation of the Gaussian probability distribution (Kocijan, 2016). It is a stochastic process f for which any finite set of values $f(\boldsymbol{z}_i)$ is jointly normally distributed For a selection of points $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M$, we label the joint probability density function of $f(\boldsymbol{z}_1), \ldots, f(\boldsymbol{z}_M)$ as

$$p(f(\boldsymbol{z}_1), \ldots, f(\boldsymbol{z}_M)) = \mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma}), \quad (1)$$

where $\boldsymbol{m}$ is the mean vector and $\boldsymbol{\Sigma}$ stands for the covariance matrix.

In GP modelling, we use a GP to map the regression vector $\boldsymbol{z}(t) = [y(t-a_1), \ldots, y(t-a_i), u(t-b_1), \ldots, u(t-b_i)]^T$ to the model output $y(t)$, where t is the time index. We construct the GP through a mean function and a covariance function. The components m_i of the mean vector $\boldsymbol{m}$ are taken to be the values of a mean function $m(\boldsymbol{z})$, $m_i = m(\boldsymbol{z}_i)$, while the matrix elements $\boldsymbol{\Sigma}_{ij}$ are the values of a covariance function $k(\boldsymbol{z}, \boldsymbol{z}')$,

$$\boldsymbol{\Sigma}_{ij} = k(\boldsymbol{z}_i, \boldsymbol{z}_j). \quad (2)$$

The role of covariance function can be served by any function that results in a positive, semi-definite covariance matrix (Kocijan, 2016).

The output of the GP model at the regression vector $\boldsymbol{z}^*$ is the probability density function $p(f(\boldsymbol{z}^*)|\mathcal{D}, \boldsymbol{z}^*)$, where the training data $\mathcal{D}$ is

$$\mathcal{D} = \{\mathbf{Z}, \boldsymbol{y}\} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N], [y_1, \ldots, y_N]^T$$

and the measured output y_i corresponds to the input $\boldsymbol{z}_i$. We assume that the training data are noisy realizations of the GP model, $f(\boldsymbol{z}_i) = y_i + \nu_i$, where the noise is uncorrelated, $\nu_i = \mathcal{N}(0, \sigma_\nu^2)$. The mean function $m(\boldsymbol{z})$ can be taken to be identically equal to 0, $m(\boldsymbol{z}) \equiv 0$. Under these assumptions, the model output $p(f(\boldsymbol{z}^*)|\mathcal{D}, \boldsymbol{z}^*)$ equals

$$p(f(\boldsymbol{z}^*)|\mathcal{D}, \boldsymbol{z}^*) = \mathcal{N}(\mu(\boldsymbol{z}^*), \sigma^2(\boldsymbol{z}^*)), \quad (3)$$

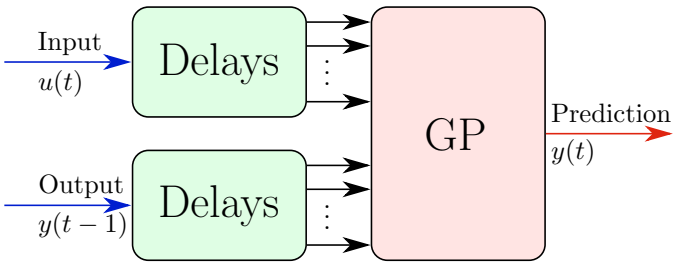

Fig. 1. The idea of GP-NARX. The regressors of the GP are delayed values of the input variables and of the output variable.

where $\mu(\boldsymbol{z}^*)$ and $\sigma^2(\boldsymbol{z}^*)$ are defined by

$$\mu(\boldsymbol{z}^*) = \boldsymbol{k}^{\mathrm{T}}\mathbf{K}^{-1}\boldsymbol{y} \quad (4)$$

$$\sigma^2(\boldsymbol{z}^*) = \kappa(\boldsymbol{z}^*) - \boldsymbol{k}^{\mathrm{T}}\mathbf{K}^{-1}\boldsymbol{k}. \quad (5)$$

The symbols $\boldsymbol{k}$ and κ stand for $\boldsymbol{k}_{1\times N} = k(\boldsymbol{z}^*, \mathbf{Z})^{\mathrm{T}}$ and $\kappa_{1\times 1} = k(\boldsymbol{z}^*, \boldsymbol{z}^*)$. The matrix $\mathbf{K}$ is defined as $\mathbf{K} = \boldsymbol{\Sigma} + {\sigma_\nu}^2\mathbf{I}$, where $\boldsymbol{\Sigma}$ is obtained using the covariance function as in (2). To obtain the probability density of the measured output y^* at $\boldsymbol{z}^*$, noise has to be taken into account. The resulting expression is

$$p(y^*|\mathcal{D}, \boldsymbol{z}^*) = \mathcal{N}(\mu(\boldsymbol{z}^*), \sigma^2(\boldsymbol{z}^*) + {\sigma_\nu}^2). \quad (6)$$

The mean function $m(\boldsymbol{z})$ can be taken to be identically equal to 0, $m(\boldsymbol{z}) \equiv 0$, while the choice of the covariance function $k(\boldsymbol{z}, \boldsymbol{z}')$ and the noise variance σ_ν^2 have to be suitable for the modelled system.

We do not have sufficient knowledge of the system to completely define the covariance function without relying on the training data. We use optimization to decide on the values of the parameters of the covariance function $\boldsymbol{\Theta}$ that are named *hyperparameters* and the noise variance from the training data. Starting from the prior assumption that every value of each hyperparameter is equally likely, the expression

$$p(\boldsymbol{\Theta}|\mathbf{Z}, \boldsymbol{y}) \propto p(\boldsymbol{y}|\mathbf{Z}, \boldsymbol{\Theta}) \quad (7)$$

follows for the likelihood $p(\boldsymbol{\Theta}|\mathbf{Z}, \boldsymbol{y})$ of the hyperparameters given the training data (Kocijan, 2016). The right-hand side of the equation is a normal distribution, the logarithm of the likelihood is (Kocijan, 2016)

$$\log p(\boldsymbol{y}|\mathbf{Z}, \boldsymbol{\Theta}) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{1}{2}\boldsymbol{y}^{\mathrm{T}}\mathbf{K}^{-1}\boldsymbol{y} \quad (8)$$

and we use this expression in choosing the values of the hyperparameters $\boldsymbol{\Theta}$ that are the most likely. We use a squared exponential covariance function with automatic relevance determination, meaning that the hyperparameters are the 15 length scales corresponding to the individual regressors. The hyperparameters and the noise variance are optimized together using the conjugate gradient method as implemented in Rasmussen and Nickisch (2010).

2.2 Model structure

We model a dynamic system with a GP in GP-NARX (nonlinear autoregressive model with exogenous input) structure, which means that the GP uses delayed output

and input values as the regressors as outlined in Fig. 1. GP-NARX is described by the equation

$$\hat{y}(t) = f(y(t-1), y(t-2) \ldots, y(t-n), u(t), u(t-1), \ldots, u(t-m)) + \nu, \quad (9)$$

where $\hat{y}$ is the prediction of output, t is the time index, f is the GP, n is the maximum lag in the output values, m is the maximum lag in the input values, and ν is Gaussian noise.

2.3 Data set description

The F-16 aircraft benchmark example data was obtained by shaking the aircraft with two dummy payloads attached. The interfaces between the aircraft and the payloads caused nonlinear distortions. The provided signals are the input voltage to the shaker under the right wing, the force of the shaker, and acceleration at 3 points on the aircraft and the payload, all sampled at 400 Hz. Multiple experiments were done with various sizes and shapes of the input signal (Noël and Schoukens, 2017).

In the study, we use the shaker force as the model input signal and the acceleration on the right wing next to the nonlinear interface as the model output signal. We use the highest excitation level multisine data set 7 for training and the next to highest excitation level data set 6 for testing. Both experiments were done with periodic forcing with the period of 8192 sampling times and 9 periods were acquired (Noël and Schoukens, 2017), resulting in 73728 data points per experiment.

2.4 Regressor selection

As the first step in data selection, we want to skip several lagged values from (9).

We begin by choosing the regressor selection method. A method based on Lipschitz quotients was tested on the same benchmark example by Perne and Stepančič (2018). 40 regressor candidates were used and 13 regressors were selected based on 14742 regression vectors from the force level 7 multisine excitation experiment. We selected the 13 most relevant regressors according to each one of 7 suitable methods implemented in ProOpter IVS (Gradišar et al., 2015) based on the same data. For each choice of regressors, we trained a GP-NARX model on 1474 regression vectors from the same experiment and tested it by prediction on the force level 6 multisine excitation data set. The best performing model in $e_{\mathrm{RMS}t}$ criterion (10) (Noël and Schoukens, 2017) was the one based on the regressors selected with linear-in-the-parameters (LIP) method (Li and Peng, 2007) as implemented in ProOpter IVS, the selected regressors being the output delayed by 1, 2, 4, 7, 8, 10, 13 time steps and the input delayed by 0, 1, 6, 8, 17, 19 steps. We thus use the LIP method to select the regressors for our study.

As the possible regressors, we use the excitation force delayed for between $b = 0$ and $b = 49$ time steps and the 2nd acceleration signal delayed for between $a = 1$ and $a = 50$ time steps, in total 100 candidates. The 15 most relevant regressors are selected using LIP based on every 5th regression vector of the force level 7 multisine excitation data set, 14736 [1] regression vectors in total. A subset of the data points is used in order to reduce the computational demands.

2.5 Data sample selection

The number of operations required for GP model training is proportional to the third power of the number of data points, $\mathcal{O}(N^3)$, restricting the number of training data points (Kocijan, 2016). The amount of training data available is too big to process, so a sample is used.

Of the several known ways of sampling the training data, the most self-evident one is decimation, that is, using every n-th training data point. The decimation factor n should be chosen so as to produce a sample of the desired size. We also take care that the greatest common divisor of the period of the input signal and n is small in order to avoid artefacts.

With advanced sample selection algorithms, care has to be taken that the algorithm itself does not require too many operations or too much computational resources, considering that the number of data points entering the selection algorithm is big.

We test the '*smart*' method for data sample selection proposed by Perne et al. (2019) and compare it with decimation. The method is based on Euclidean distance between training data points. Every data point from the training data set is treated as a vector with normalized regressor and output values as its coordinates. Euclidean distances between all pairs of data points are computed. The points closest to their nearest neighbours are rejected. The procedure is done iteratively, 5 % of the points are discarded in every cycle until the desired number of training points is reached. When comparing the methods, we ensure that the number of training data points is the same between them.

There are multiple reasons for comparing these two sample selection methods. Decimation is conceptually clear and fast to both implement and compute. Of the more advanced methods, the 'smart' method is sufficiently fast and easily available as it is easy to program, so it is the perfect method to try out to observe the effect of data sample selection. Most other methods would require a significant amount of programming effort or computational resources, or both.

2.6 Model evaluation

To evaluate the models, we compare the model predictions to the test data set, obtaining the relevant figures of merit.

The model prediction is done for the time steps for which the regressor values are available from the measurements. The predicted normal distribution of the system output is calculated from (6) and given as the predicted mean and the predicted variance at every time step.

The benchmark figure of merit for the benchmark example $e_{\mathrm{RMS}t}$ prescribed by Noël and Schoukens (2017) is calcu-

[1] The number of available regression vectors is smaller than the number of measurements as not all the delayed values are available for the first 50 measurements.

lated from the predicted mean and the measured system output by the formula

$$e_{\mathrm{RMS}t} = \sqrt{N^{-1}\sum_{t=1}^{N}(y_t-\mu)^2}, \tag{10}$$

where y_t is the measured output, μ the model predicted mean value, and N the number of prediction points.

Since the $e_{\mathrm{RMS}t}$ figure of merit does not depend on the predicted variance while the main benefit of GP modelling is variance prediction, there is a need for another figure of merit. We choose the mean standardised log loss (MSLL), (Rasmussen and Williams, 2006, p. 23)

$$\mathrm{MSLL} = \frac{1}{2N}\sum_{t=1}^{N}\left[\ln\left(\sigma^2\right) - \ln\left(\sigma_y^2\right) + \frac{\left(\mu-y_t\right)^2}{\sigma^2} - \frac{\left(y_t - E\left(\boldsymbol{y}\right)\right)^2}{\sigma_y^2}\right], \tag{11}$$

where $E(\boldsymbol{y})$ is the mean of the measured value, σ_y^2 is the variance of the measured value, μ is the mean prediction, and σ^2 is the predictive variance. MSLL is zero for a model returning the sample mean and sample variance of the test outputs and smaller for better models (Chen and Wang, 2018).

We are thus using two different figures of merit and calculating them for the same model runs. We treat them completely separately, drawing conclusions from the calculated $e_{\mathrm{RMS}t}$ results independently from the conclusions based on the MSLL values. Which figure of merit is appropriate or more relevant depends on the model purpose. Investigating two figures of merit is thus equivalent to investigating two different model uses.

3. RESULTS

The 15 selected regressors are the value of the input variable delayed by 1, 4, 8 time steps and the value of the output variable delayed by 1, 2, 4, 7, 8, 10, 14, 17, 21, 24, 28, 32 time steps.

Ten models are constructed based on the selected regressors and different training data samples. Two of them use 4913 data points selected either through decimation or by the 'smart' method. The number of 4913 training data points results from decimating with the decimation factor of 15 and is reasonably close to the upper limit of our computer system. We require the same number of data points from the 'smart' method as well so that we can compare the effect of a change in training data sample quality without a change in quantity. To observe the consequences of a change in the training data sample size, we multiply the decimation factor by 10 to obtain the sample size of 491 and also select 491 training data points with the 'smart' method. Decimation factors of 300, 500, and 1000 are also used, and a model based on the same number of 'smartly' selected training data points is produced for each of these' too.

The model prediction for 2000 time steps of the test data set by the model with 491 'smartly' selected training data points is shown in Fig. 2. For comparison, there is the model prediction for the same time period based on the

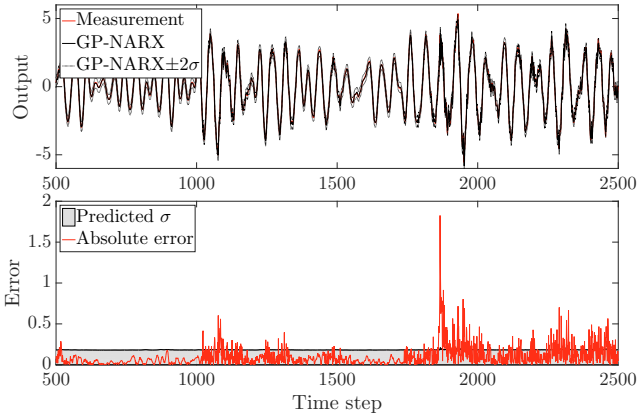

Fig. 2. Prediction of the GP-NARX model with 491 'smartly' selected training data points

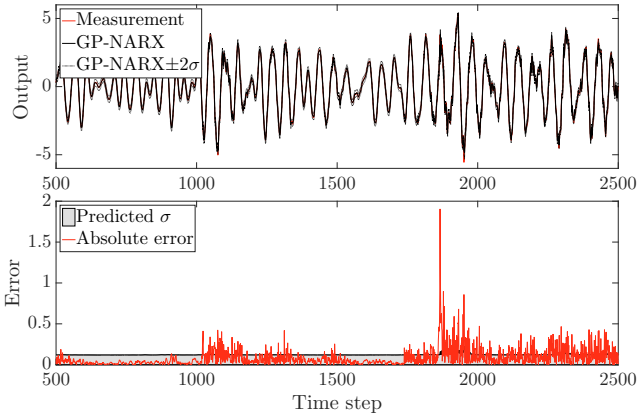

Fig. 3. Prediction of the GP-NARX model with 4913 training data points selected with decimation

model with 4913 decimated training data points in Fig. 3. The figures of merit are calculated for predictions of all 4 models and are presented in Fig. 4 and Table 1.

It is interesting to observe the mean value of the third term in the sum (11) that we call r,

$$r = \frac{1}{N}\sum_{t=1}^{N}\frac{\left(\mu-y_t\right)^2}{\sigma^2},$$

we provide it in Table 1. This term is one of the two terms in the sum that depend on the model prediction, the other one is the first term $\ln\left(\sigma^2\right)$. In Table 1 we also provide the average $\overline{\sigma^2} = \frac{1}{N}\sum_{t=1}^{N}\sigma^2$ which is related to the first term in the sum (11). It should be noted that the quantity r is related to $e_{\mathrm{RMS}t}$.

We see in Table 1 that the 'smart' selection improves $e_{\mathrm{RMS}t}$ and worsens MSLL compared to decimation, except for 73 data points, where 'smart' selection improves both. The $e_{\mathrm{RMS}t}$ figure of merit is based on the prediction of the mean value, while MSLL also takes into account the predicted variance σ^2. MSLL can worsen when $e_{\mathrm{RMS}t}$ improves only if the prediction of variance worsens, so we can infer that the prediction of σ^2 typically worsens with 'smart' sampling. The change in predicted variance is systematic and results in an increase of the average $\overline{\sigma^2}$. As a result of both increase in the average predicted variance and decrease in mean square error, r decreases. If

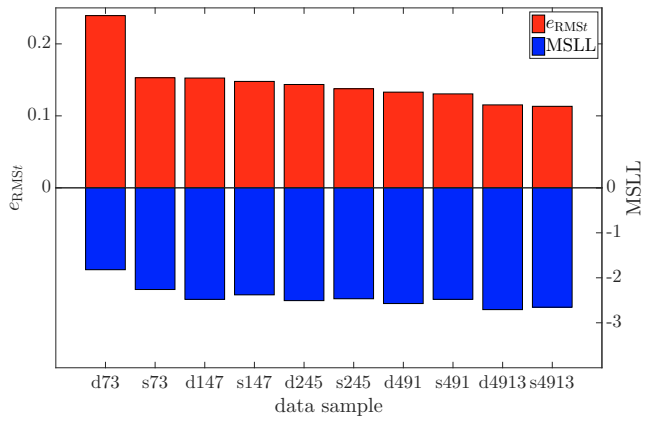

Fig. 4. Figures of merit for predictions with models based on different training data samples

σ^2 was a constant independent of t, which is a reasonable approximation considering that the variance of predicted variance $\mathrm{Var}\left(\sigma^2\right)$ is small, MSLL would be smallest if σ^2 was chosen so that $r = 1$. We see that $r < 1$ for most models, they are biased towards high values of σ^2. Except at 73 data points, the decrease in r by the 'smart' selection is more than compensated for by the increase in the first term of (11) resulting in the observed increase in MSLL.

To summarise, most models have their average predicted variance $\overline{\sigma^2}$ bigger than optimal, meaning that they have too little confidence in their predictions. As the ones with the 'smart' data sampling are the most lacking in confidence, their MSLL is worse even though their mean prediction is better, except in the case of 73 training points.

4. DISCUSSION

The increase in the number of training data points improves the model predictions in both $e_{\mathrm{RMS}t}$ and MSLL measures. This behaviour is expected.

Except at 73 training data points, where 'smart' selection clearly outperforms decimation, the effects of different training data sampling methods are contradictory. 'Smart' selection of training data points improves the mean prediction compared to decimation as seen from the $e_{\mathrm{RMS}t}$ values. On the contrary, MSLL value is better if the training data points are sampled using decimation than if 'smart' selection is used. The finding holds true for the models

Table 1. Figures of merit and some other statistics of predictions with models based on different training data samples

Data sample choice	N	$e_{\mathrm{RMS}t}$	MSLL	r	$\overline{\sigma^2}$	$\mathrm{Var}\left(\sigma^2\right)$
decimation	73	0.239	−1.82	2.59	0.017	$1 \cdot 10^{-4}$
'smart'	73	0.153	−2.26	0.40	0.057	$1 \cdot 10^{-5}$
decimation	147	0.153	−2.48	1.44	0.014	$3 \cdot 10^{-5}$
'smart'	147	0.148	−2.38	0.55	0.038	$2 \cdot 10^{-6}$
decimation	245	0.144	−2.51	0.97	0.020	$5 \cdot 10^{-6}$
'smart'	245	0.138	−2.47	0.61	0.030	$1 \cdot 10^{-6}$
decimation	491	0.133	−2.57	0.94	0.018	$2 \cdot 10^{-6}$
'smart'	491	0.131	−2.48	0.50	0.033	$1 \cdot 10^{-6}$
decimation	4913	0.115	−2.71	0.86	0.015	$2 \cdot 10^{-6}$
'smart'	4913	0.113	−2.65	0.58	0.021	$1 \cdot 10^{-6}$

with both the smaller and the bigger number of training data points.

It is thus not clear which model is better – the one based on decimated or the one based on 'smartly' selected training data points. How good a model is should be measured with respect to the model objectives (Ljung, 1999), and the figures of merit used are supposed to reflect those objectives. The model that is better in $e_{\mathrm{RMS}t}$ is better for the purposes where one only needs the predicted mean value, such as where one's goal is to be able to predict an acceleration that will be as close to the observed acceleration as possible. The model with the better MSLL value is better for other purposes, for example, when one wants to have a prediction and at the same time know how much trust to put into the prediction. We have encountered an example where a different choice of the figure of merit propagates all the way to a different data sampling method being favoured. The two different data sampling methods result in models giving considerably different predictions and we cannot give a general answer on which one performs better, even when both are tested on the same data set.

The results demonstrate that if data sample selection is used, it has to be done carefully. The intention is to pick a 'better' sample and get a 'better' model. Data sample selection is thus meant to change the model. In the extreme case, one could choose the desired model parameters and select the sample based on them. It is clear that such cherry picking of data is to be avoided – however, if the sample is not directly selected for the desired model parameters, it does not guarantee that there will not be trouble. Since the figure of merit is used in the definition of what it means for a model to be 'better', the figure of merit may influence which data set selection method is better. The studied system, data sample selection methods, and figures of merit offer such an example.

5. CONCLUSION

We use the F-16 ground vibration test benchmark data set to test a computationally efficient way of data sampling based on Euclidean distance and compare it with decimation. The intended purpose of the 'smart' data sampling method is modelling, therefore we use the data samples to train models. We test the models on a data set separate from the training data set and compare the figures of merit.

The results show that the research question was ill-posed. Training data sampling methods cannot be ranked from best to worst any more than models can be ranked. Models can be ranked in suitability for a particular purpose – and the purported use of the model percolates to the other side and determines the suitability of the data sample. In particular, decimation is better when we want a good MSLL and 'smart' sampling is better for $e_{\mathrm{RMS}t}$, at least with the system studied. We want to emphasise that we do not vary the test data set. The change of the figure of merit is sufficient to change which model better predicts the outcome of the same experiments and which way of training data sampling leads to a better model.

Data sample selection has a particularly strong influence on the predicted variance compared to the predicted mean value. This is not surprising: the sample variance is typically easier to influence than the sample mean by carefully choosing the sample.

REFERENCES

Chen, Z. and Wang, B. (2018). How priors of initial hyperparameters affect Gaussian process regression models. *Neurocomputing*, 275, 1702–1710. doi:10.1016/j.neucom.2017.10.028.

Gradišar, D., Glavan, M., Strmčnik, S., and Mušič, G. (2015). ProOpter: An advanced platform for production analysis and optimization. *Computers in Industry*, 70, 102–115. doi:10.1016/j.compind.2015.02.010.

Khosravani, H., Ruano, A., and Ferreira, P. (2016). A convex hull-based data selection method for data driven models. *Applied Soft Computing*, 47, 515–533. doi:10.1016/j.asoc.2016.06.014.

Kocijan, J. (2016). *Modelling and Control of Dynamic Systems Using Gaussian Process Models*. Advances in Industrial Control. Springer International Publishing. doi:10.1007/978-3-319-21021-6.

Li, K. and Peng, J.X. (2007). Neural input selection–A fast model-based approach. *Neurocomputing*, 70(4), 762–769. doi:10.1016/j.neucom.2006.10.011.

Lin, W.C., Tsai, C.F., Ke, S.W., Hung, C.W., and Eberle, W. (2015). Learning to detect representative data for large scale instance selection. *Journal of Systems and Software*, 106, 1–8. doi:10.1016/j.jss.2015.04.038.

Ljung, L. (1999). *System identification (2nd ed.): theory for the user*. Prentice Hall PTR, Upper Saddle River.

Naghizadeh, M. and Sacchi, M.D. (2010). On sampling functions and Fourier reconstruction methods. *Geophysics*, 75(6), WB137–WB151. doi:10.1190/1.3503577.

Noël, J. and Schoukens, M. (2017). F-16 aircraft benchmark based on ground vibration test data. In *2017 Workshop on Nonlinear System Identification Benchmarks*, 19–23. Brussels, Belgium, April 24–26, 2017. URL `http://nonlinearbenchmark.org/FILES/BenchmarkWorkshop2017_Abstracts.pdf`.

Perne, M. and Stepančič, M. (2018). Regressor selection using Lipschitz quotients on the F-16 aircraft benchmark. In *2018 Workshop on Nonlinear System Identification Benchmarks*, 18. Liège, Belgium, April 11–13, 2018. URL `http://nonlinearbenchmark.org/FILES/BenchmarkWorkshop2018_Abstracts.pdf`.

Perne, M., Stepančič, M., and Grašič, B. (2019). Handling big datasets in Gaussian processes for statistical wind vector prediction. In *5th IFAC Conference on Intelligent Control and Automation Sciences*, 121–126. Belfast, Northern Ireland, 21–23 August 2019. doi:10.1016/j.ifacol.2019.09.126.

Rasmussen, C.E. and Nickisch, H. (2010). *Gaussian Process Regression and Classification Toolbox version 3.1 for GNU Octave 3.2.x and Matlab 7.x*.

Rasmussen, C.E. and Williams, C.K.I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Schoukens, M. and Noël, J.P. (2015). Benchmark nonlinear system identification: Benchmarks. URL `http://nonlinearbenchmark.org/`. Accessed: 2019-09-25.

Silva, D.A., Souza, L.C., and Motta, G.H. (2016). An instance selection method for large datasets based on Markov Geometric Diffusion. *Data & Knowledge Engineering*, 101, 24–41. doi:10.1016/j.datak.2015.11.002.

Tang, T., Chen, S., Zhao, M., Huang, W., and Luo, J. (2019). Very large-scale data classification based on K-means clustering and multi-kernel SVM. *Soft Computing*, 23(11), 3793–3801. doi:10.1007/s00500-018-3041-0.

Wang, K., Pleiss, G., Gardner, J., Tyree, S., Weinberger, K.Q., and Wilson, A.G. (2019). Exact Gaussian processes on a million data points. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32, 14648–14659. Curran Associates, Inc. URL `https://papers.nips.cc/paper/9606-exact-gaussian-processes-on-a-million-data-points`.