Application of on-line Gaussian process models for pressure signal

D. Petelin[1], J. Kocijan[1,2]

[1] Jožef Stefan Institute, Department of Systems and Control, Ljubljana, Slovenia
[2] University of Nova Gorica, Nova Gorica, Slovenia

I. Introduction

Gaussian process (GP) models form a new, emerging complementary method for nonlinear system identification. GP model is a probabilistic nonparametric black-box model. It differs from most of the other frequently used black-box identification approaches (such as Neural Networks [1]) as it does not try to approximate the modelled system by fitting the parameters of the selected basis functions but rather searches for the relationship among measured data. Because GP model is a Bayesian model, the output of Gaussian process model is a normal distribution, expressed in terms of mean and variance. Mean value represents the most likely output and the variance can be viewed as the measure of its confidence. Obtained variance, which depends on amount of available identification data, is important information distinguishing the GP models from other non-Bayesian methods. Gaussian process can be used for model identification when data are heavily corrupted with noise, when there are outliers or gaps in the input data. Another useful attribute of GP model is the possibility to include various kinds of prior knowledge into the model, e.g. local models, static characteristic, etc.

A noticeable drawback of the system identification with Gaussian process models is computation time necessary for modelling. Gaussian process regression involves several matrix computations which load increases with the third power of the number of input data, such as matrix inversion and the calculation of the log-determinant of used covariance matrix. This computational greed restrict the number of training data, to at most a few thousand cases.

To overcome the computational limitation issues and make use of the method also for large-scale dataset application, numerous authors have suggested various sparse approximations. Authors of [2] have provided a unified view of sparse Gaussian process approximation, which includes a comparison of work published by various authors. Common to all these approximation methods is that only a subset of the variables is treated exactly, with the remaining variables given some approximate, but computationally cheaper approach. On-line learning of Gaussian process models can be treated as the special case of sparse Gaussian process approximation. Such method, that is in the focus of this paper, is sparse on-line Gaussian processes learning method [3].

The purpose of this paper is to make a case study of using an on-line sparse Gaussian processes learning method for modelling a pressure signal of gas-liquid separation process. Obtained model would be used for the one-step-ahead prediction for the development or operation of the system.

The paper is composed as follows. The next section will briefly describe the modelling of dynamic systems with Gaussian process models. The case study will follow in the third section and the conclusions are given at the end of the paper.

II. Modelling of dynamic systems with Gaussian processes

A Gaussian process model is a flexible, probabilistic, non-parametric model with uncertainty predictions. Its uses and properties for modelling are reviewed in [4]. The use of Gaussian processes for modelling dynamic systems is a relatively recent development [5]. A retrospective review can be found

ISBN 978-615-5044-00-7

in [6].

A Gaussian process is a collection of random variables that have a joint multivariate Gaussian distribution. The mean $\mu(\mathbf{x})$ and the covariance function $C(\mathbf{x}_p, \mathbf{x}_q)$ fully specify the Gaussian process. Note that the covariance function $C(.,.)$ can be any function that has the property of generating a positive semidefinite covariance matrix.

The covariance function $C(\mathbf{x}_p, \mathbf{x}_q)$ can be interpreted as a measure of the distance between the input points $\mathbf{x}_p$ and $\mathbf{x}_q$. For systems modelling it is usually composed of two main parts, representing the functional part and the noise part.

A common choice is

$$C_f(\mathbf{x}_p, \mathbf{x}_q) = v_1 \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d(x_{dp} - x_{dq})^2\right] + \delta_{pq} v_0, \tag{1}$$

where $\mathbf{\Theta} = [w_1 \dots w_D \; v_0 \; v_1]^T$ are the 'hyperparameters' of the covariance functions, D is the input dimension and $\delta_{pq} = 1$ if $p = q$ and 0 otherwise. Other possible covariance functions are given in [4]. The square exponential covariance function represents smooth and continuous functional part and the constant covariance function represents the noise part when it is presumed to be the white noise. For a given problem, the parameters are learned or identified using the data at hand.

Consider a set of N D-dimensional input vectors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$ and a vector of output data $\mathbf{y} = [y_1, y_2, \dots, y_N]^T$. Based on the data $(\mathbf{X}, \mathbf{y})$, and given a new input vector $\mathbf{x}^*$, we wish to find the predictive distribution of the corresponding output y^*. Unlike other models, there is no model-parameter determination as such, within a fixed model structure. With this model, most of the effort involves *tuning* the parameters of the covariance function. This is done by maximising the log marginal likelihood ($\log(p(\mathbf{y}|\mathbf{X})) = -\frac{1}{2}\log(\mid \mathbf{K} \mid) - \frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\mathbf{y} - \frac{N}{2}\log(2\pi)$)), where $\mathbf{K}$ is the $N \times N$ training covariance matrix. The number of parameters to be optimized is small ($D + 2$, see equation (1)), which means that the optimization convergence might be faster and that the 'curse of dimensionality' so common to black-box identification methods is circumvented or at least decreased.

The described approach can be easily utilized for regression calculations. Based on the training set $\mathbf{X}$ a covariance matrix $\mathbf{K}$ of size $N \times N$ is determined. As already mentioned, the aim is to find the distribution of the corresponding output y^* at some new input vector $\mathbf{x}^* = [x_1(N+1), x_2(N+1), \dots, x_D(N+1)]^T$.

For a new test input $\mathbf{x}^*$, the predictive distribution of the corresponding output is $y^*|(\mathbf{X}, \mathbf{y}), \mathbf{x}^*$ and this is Gaussian, with a mean and a variance

$$\mu(\mathbf{x}^*) = \mathbf{k}(\mathbf{x}^*)^T \mathbf{K}^{-1} \mathbf{y}, \tag{2}$$

$$\sigma^2(\mathbf{x}^*) = k(\mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*)^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}^*), \tag{3}$$

where $\mathbf{k}(\mathbf{x}^*) = [C(\mathbf{x}^1, \mathbf{x}^*), \dots, C(\mathbf{x}^N, \mathbf{x}^*)]^T$ is the $N \times 1$ vector of covariances between the test and training cases, and $k(\mathbf{x}^*) = C(\mathbf{x}^*, \mathbf{x}^*)$ is the covariance between the test input and itself [4].

Gaussian processes can, like other machine learning methods, e.g. neural networks, be used to model static nonlinearities and can therefore be used for the modelling of dynamic systems [7] if the delayed input and output signals are fed back and used as regressors. In such cases an autoregressive model is considered, such that the current output depends on the previous outputs, as well as on the previous control inputs.

$$\begin{aligned} \mathbf{x}(k) &= [y(k-1), y(k-2), \dots, y(k-L), \\ &\quad u(k-1), u(k-2), \dots, u(k-L)]^T, \\ y(k) &= f(\mathbf{x}(k)) + \epsilon, \end{aligned} \tag{4}$$

where k denotes the consecutive number of the data sample. Let $\mathbf{x}$ denote the state vector composed of the previous outputs y and inputs u up to a given lag L, and ϵ is white noise.

ISBN 978-615-5044-00-7

As can be seen from the presented relations, the obtained model not only describes the dynamic characteristics of the nonlinear system, but also provides information about the confidence in these predictions by means of the prediction variance. The Gaussian process can highlight areas of the input space where the prediction quality is poor, due to the lack of data, by indicating a higher variance around the predicted mean.

1. On-line modelling

A noticeable drawback of system identification with Gaussian process models is the computation time necessary for the modelling. Gaussian process regression involves several matrix computations in which the load increases with the third power of the number of input data, such as matrix inversion and the calculation of the log-determinant of the used covariance matrix. This computational greed restricts the amount of training data, to at most a few thousand cases. To overcome the computational-limitation issues and to also make use of the method for large-scale dataset applications, numerous authors have suggested various sparse approximations [2, 8] as well as on-line modelling [3], which is a special kind of sparse approximate method. A common property to all sparse approximate methods is that they try to retain the bulk of the information contained in the full training dataset, but reduce the size of the resultant covariance matrix so as to facilitate a less computationally demanding implementation of the GP model.

The selected on-line learning method [9, 3, 10], suited to our problem, is based on a combination of a Bayesian on-line approach [11] and a sequential construction of a relevant sub-sample of the data on which an approximation of the GP model is based. This approximation is obtained by using parametrisation and projection techniques. To keep the subset of the most relevant data a fixed size there are two types of update to the GP model: a *basic* update that is performed when the error of a new approximation is smaller than a defined threshold, and a *full* update, which is performed otherwise. While a basic update only updates parameters that present the approximation, without increasing their number, a full update, besides updating parameters, also adds current data to the subset of the most relevant data. If this operation results in the maximum size of the subset being exceeded, the least relevant data is removed.

III. Case study

The semi-industrial process plant used for the case study in this paper is the unit for separating the gas from the liquid that forms part of a larger pilot plant.

The role of the separation unit is to capture the flue gases under low pressure from the effluent channels by means of a water flow, to cool them down and then supply them under high-enough pressure to other parts of the pilot plant.

The flue gases coming from the effluent channels are absorbed by the water flow into the water circulation pipe through the injector.

The water flow is generated by the water ring pump. The speed of the pump is kept constant. The pump feeds the mixture of water and gas into the tank, where the gas is separated from the water. Hence, the accumulated gas in the tank forms a sort of 'gas cushion' with an increased internal pressure. Owing to this pressure, the flue gas is blown out from the tank into the neutralization unit. On the other hand, the 'cushion' forces the water to circulate back to the reservoir. The quantity of water in the circuit is constant.

Since the process to be identified is characterised as predominantly a first-order system a model of the form 5 is identified

$$p_1(k+1) = f(p_1(k), u_1(k), h_1(k)). \tag{5}$$

This means that the pressure $p_1(k)$, the valve signal $u_1(k)$ and the liquid level $h_1(k)$ are selected for regressors. Ref. [12] contains a more elaborate study on the choice of model order and regressors for

ISBN 978-615-5044-00-7

this process.

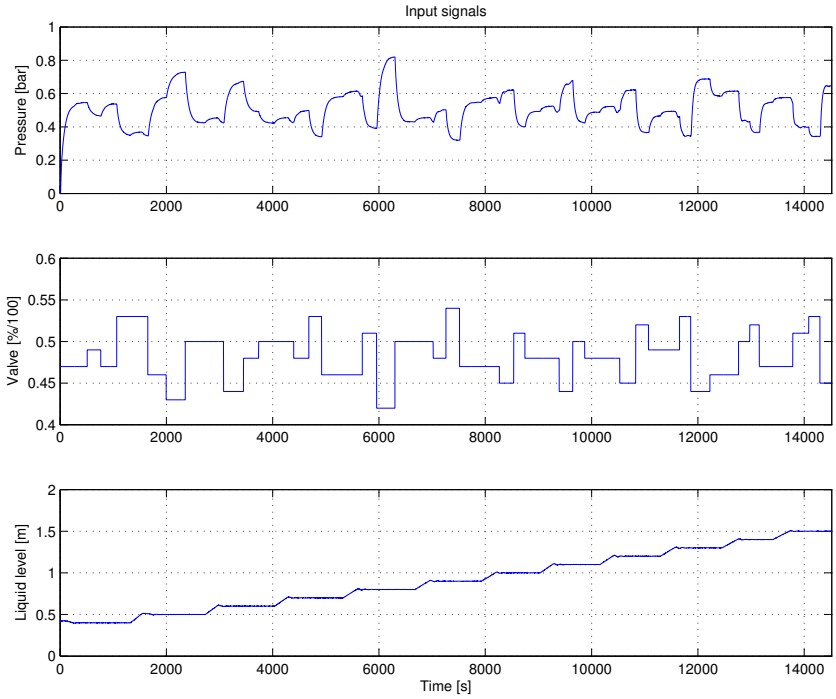

Figure 1: Signals that are used for regressors of the identified model

For training were used signals, given in Figure 1, which were obtained from experiment on the plant described before. The experiment was running 4 hours with measurement interval of 1 second. The measured values of pressure are between 0.4 and 0.7 bar, of liquid level are between 0.4 and 1.5 meters, of valve aperture are between 0 (close) and 1 (open).

While the computational complexity of an on-line Gaussian process model raises only linearly with the amount of training data, all available data (14520) was used for training. As it was already mentioned, the on-line Gaussian process model is based on the set of basis vectors, which best describes the process. In order to get as best as possible model, the maximum size of the set of basis vectors was limited to approximately the tenth of all data - that is 1500 samples. However, the obtained model finally contains 1191 basis vectors.

The distribution of basis vectors through the whole process can be seen from Figure 2, where basis vectors are marked with red crosses. These evenly distributed basis vectors enable accurate one-step-ahead predictions. The accuracy of one-step-ahead predictions are validated with four error measures:

- mean squared error: $MSE = \frac{1}{n}\sum_{i=1}^{n}(y(i) - \hat{y}(i))^2$,
- mean absolute error: $MAE = \frac{1}{n}\sum_{i=1}^{n}|y(i) - \hat{y}(i)|$,
- minus log-predicted density error: $LPD = \frac{1}{2n}\sum_{i=1}^{n}\left(\log(2\pi) + \log(\sigma) + \frac{(y(i)-\hat{y}(i))^2}{\sigma}\right)$,
- mean relative square error: $MRSE = \sqrt{\frac{\sum_{i=1}^{n}(y(i)-\hat{y}(i))^2}{\sum_{i=1}^{n}y(i)^2}}$.

All these error measures for obtained model are given in Table 1. It is clear that the one-step-ahead predictions are sufficiently accurate through the whole process. In conclusion, obtained model is useful for any task where a short-time-prediction for the development or operation of the system is needed.

ISBN 978-615-5044-00-7

MSE	MAE	LPD	MRSE
0,0001461	0,0073	-0,5902	0,1201

Table 1: The table of different error measures for validating the model prediction

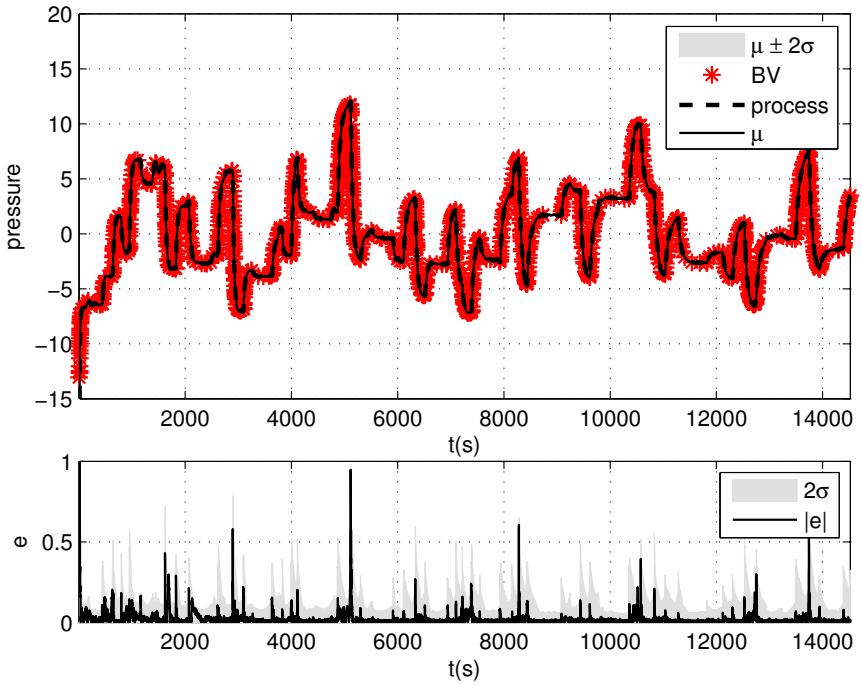

Figure 2: One-step-ahead prediction with Gaussian process model (top) with corresponding confidence band and absolute values of prediction error (bottom)

We also wanted to find out the error measures dependency of the amount of basis vectors. Therefore we sequentially removed the worst scored basis vector from the set of basis vectors and validated the obtained model with previously described error measures. The results are given in Figure 3.

It can be seen that error measures stabilise by different amount of basis vectors. Error measures that not take into account the variance of prediction (MSE, MAE, MRSE) stabilise much faster (approximately by 300 basis vectors) than LPD. Latter takes into account variance and is therefore more suitable for validating Bayesian models. It stabilises by approximately 600 basis vectors. That means the process containing 14520 samples could be sufficiently described with 600 basis vectors.

IV. Conclusion

An on-line learning of Gaussian process models is proposed in this paper for one-step-ahead predictions of pressure signal used for any task where a short-time-prediction for the development or operation of the system is needed. The obtained results in the case study show that the on-line learning algorithm evenly distributes the basis vectors through the whole process and therefore makes the one-step-ahead predictions sufficiently accurate.

Future work will be directed towards an improvement of the model for multi-step-ahead predictions and validation with simulation. The success of simulation would be good basis for predictive control of this process in practice.

ISBN 978-615-5044-00-7

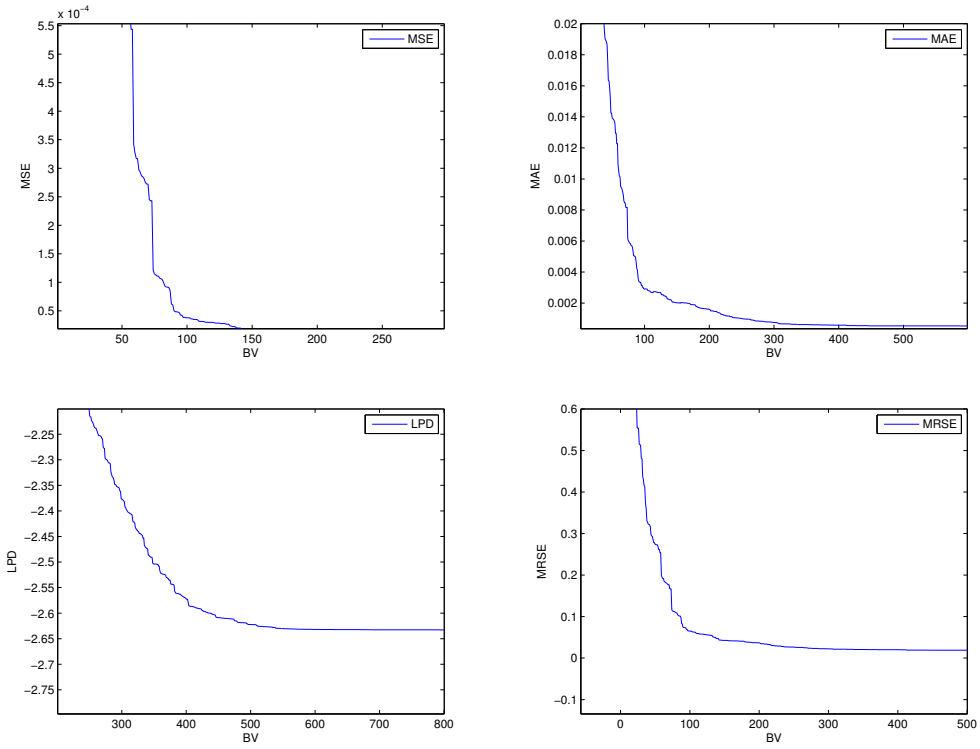

Figure 3: Comparison of used error measures depending from the number of utilized basis vectors

Acknowledgement

This work was financed by the Slovenian Research Agency, grants Nos. P2-0001 and J2-2099. Authors are grateful to prof. Janez Demšar for useful discussions about the paper topic.

References

[1] J. Sjöberg, *Non-Linear System Identification with Neural Networks*, Ph.D. thesis, Linköping, 1995.

[2] J. Quinonero-Candela, "A unifying view of sparse approximate Gaussian process regression," *J. Mach. Learn. Res.*, 6:1939–1959, 2005.

[3] L. Csató and M. Opper, "Sparse on-line Gaussian processes," *Neural Comput.*, 14(3):641–668, 2002.

[4] C. E. Rasmusen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA, 2006.

[5] K. Ažman and J. Kocijan, "Application of Gaussian processes for black-box modelling of biosystems," *ISA Transactions*, 46(4):443–457, 2007.

[6] J. Kocijan, "Gaussian process models for systems identification," *Proceedings of the 9-th International PhD Workshop on Systems and Control: young generation viewpoint*, p. 8, 2008.

[7] J. Kocijan, A. Girard, B. Banko, and R. Murray-Smith, "Dynamic systems identification with Gaussian procesess," *Mathematical and Computer Modelling of Dynamic Systems*, 11(4):411–424, 2005.

[8] J. Quinonero-Candela, C. E. Rassmusen, and C. K. I. Williams, "Approximation methods for Gaussian process regression," Tech. Rep., Microsoft Research, September 2007.

[9] L. Csató and M. Opper, "Sparse representation for gaussian process models," *Advances in neural information processing systems*, 13:444–450, 2001.

[10] L. Csató, *Gaussian Processes - Iterative Sparse Approximation*, Ph.D. thesis, Aston University, 2002.

[11] M. Opper, *A Bayesian approach to on-line learning*, pp. 363–378, Cambridge University Press, 1998.

[12] A. Girard, *Approximate methods for propagation of uncertainty withgaussian process models*, Ph.D. thesis, University of Glasgow, Glasgow, UK, 2004.

ISBN 978-615-5044-00-7