Dynamic GP models: an overview and recent developments

JUŠ KOCIJAN
Jozef Stefan Institute
Jamova 39, Ljubljana
SLOVENIA
&
University of Nova Gorica
Vipavska 13, Nova Gorica
SLOVENIA
jus.kocijan@ijs.si

Abstract: Various methods can be used for nonlinear, dynamic-system identification and Gaussian process (GP) model is a relatively recent one. The GP model is an example of a probabilistic, nonparametric model with uncertainty predictions. It possesses several interesting features like model predictions contain the measure of confidence. Further, the model has a small number of training parameters, a facilitated structure determination and different possibilities of including prior knowledge about the modelled system. The framework for the identification of dynamic systems with GP models are presented and an overview of recent advances in the research of dynamic-system identification with GP models and its applications are given.

Key–Words: Nonlinear-system identification, Gaussian process models, dynamic systems, regression, control systems, fault detection, Bayesian filtering.

1 Introduction

This paper provides an overview of recent developments and applications of Gaussian process (GP) models for experimental modelling of dynamic systems.

GP models provide a probabilistic, nonparametric modelling approach for black-box identification of nonlinear dynamic systems. They can highlight areas of the input space where prediction quality is poor, due to the lack of data or its complexity, by indicating the higher variance around the predicted mean. GP models contain noticeably less coefficients to be optimised than commonly used parametric models. This approach to modelling is not considered as a replacement to any existing system-identification method, but rather as a complementary approach to modelling. The drawback of GP models is their considerable computational burden. This burden may be perceived as an obstacle for GP models in dynamic-systems modelling.

The structure of the paper is as follows. Principles of GPaussian process modelling are briefly described in the next section. Survey of recent advances of dynamic GP models is given in the third section. Recent applications of dynamic GP models are presented in the fourth section. Trends, challenges and research opportunities are given in the fifth section.

2 Systems modelling with Gaussian processes

A GP model is a probabilistic, nonparametric model for the prediction of output-variable distributions. Its use and properties for modelling are thoroughly described in [1]. Here, only a brief description is given.

A Gaussian process is a collection of random variables which have a joint multivariate Gaussian distribution. Assuming a relationship of the form $y = f(\mathbf{x})$ between input $\mathbf{x}$ and output y, we have $y_1, \ldots, y_n \sim \mathcal{N}(0, \mathbf{\Sigma})$, where elements of the covariance matrix Σ, namely, $\Sigma_{pq} = \mathrm{Cov}(y_p, y_q) = C(\mathbf{x}_p, \mathbf{x}_q)$ give the covariance between output points corresponding to input points $\mathbf{x}_p$ and $\mathbf{x}_q$. Thus, the mean $\mu(\mathbf{x})$ and the covariance function $C(\mathbf{x}_p, \mathbf{x}_q)$ fully specify the GP.

The value of covariance function $C(\mathbf{x}_p, \mathbf{x}_q)$ expresses the correlation between the individual outputs $f(\mathbf{x}_p)$ and $f(\mathbf{x}_q)$ with respect to inputs $\mathbf{x}_p$ and $\mathbf{x}_q$. Note that the covariance function $C(\cdot, \cdot)$ can be any function that generates a positive semi-definite covariance matrix. It is usually composed of two parts:

$$C(\mathbf{x}_p, \mathbf{x}_q) = C_f(\mathbf{x}_p, \mathbf{x}_q) + C_n(\mathbf{x}_p, \mathbf{x}_q), \qquad (1)$$

where C_f represents the functional part and describes the unknown system we are modelling, and C_n represents the noise part and describes the model of the noise.

A frequently chosen covariance function consists of the square exponential covariance function for functional part C_f and the constant covariance function for the noise part C_n, and is of the following form:

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

where w_d, v_0, v_1 are the 'hyperparameters' of the covariance function, D is the dimension of input space, and $\delta_{pq} = 1$ if $p = q$ and 0 otherwise. The hyperparameters can be written as a vector $\mathbf{\Theta} = [w_1 \ldots w_D \ v_0 \ v_1]^T$. This covariance function is smooth and continuous. It presumes that the process is stationary and that the noise is white. Other forms and combinations of covariance functions suitable for various applications can be found in [1]. For a given problem, the hyperparameter values are learned using the data at hand.

To accurately reflect the correlations present in the training data, the hyperparameters of the covariance function need to be optimised. Due to the probabilistic nature of the GP models, a probabilistic approach to the optimisation of the model is used. Instead of minimizing the model error, the log of marginal likelihood is maximized.

GP models can be easily utilized for regression calculation. Consider a set of N D-dimensional input vectors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ and a vector of output data $\mathbf{y} = [y_1, y_2, \ldots, y_N]^T$. 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^*. Based on training set $\mathbf{X}$, a covariance matrix $\mathbf{K}$ of size $N \times N$ is determined. The overall problem of learning unknown parameters from data corresponds to the predictive distribution $p(y^*|\mathbf{y}, \mathbf{X}, \mathbf{x}^*)$ of the new target y, given the training data $(\mathbf{y}, \mathbf{X})$ and a new input $\mathbf{x}^*$. In order to calculate this posterior distribution, a prior distribution over the hyperparameters $p(\mathbf{\Theta}|\mathbf{y}, \mathbf{X})$ can first be defined, followed by the integration of the model over the hyperparameters

$$p(y^*|\mathbf{y}, \mathbf{X}, \mathbf{x}^*) = \int p(y^*|\mathbf{\Theta}, \mathbf{y}, \mathbf{X}, \mathbf{x}^*) p(\mathbf{\Theta}|\mathbf{y}, \mathbf{X}) d\mathbf{\Theta}. \tag{3}$$

The computation of such integrals can be difficult due to the intractable nature of the nonlinear functions. A solution to the problem of intractable integrals is to adopt numerical integration methods such as the Monte-Carlo approach. Unfortunately, significant computational efforts may be required to achieve a sufficiently accurate approximation.

An alternative approach based on the Maximum Likelihood optimisation method has been developed and is applied to maximize the marginal likelihood. It can be restated as a cost function that is to be maximized. For numerical scaling purposes the log of the marginal likelihood is taken:

$$L(\mathbf{\Theta}) = -\frac{1}{2}\log(|\mathbf{K}|) - \frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\mathbf{y} - \frac{N}{2}\log(2\pi). \tag{4}$$

The predictive distribution of the GP model output for a new test input $\mathbf{x}^*$ has normal probability distribution with mean and variance

$$\mu(y^*) = \mathbf{k}(\mathbf{x}^*)^T\mathbf{K}^{-1}\mathbf{y}, \tag{5}$$

$$\sigma^2(y^*) = \kappa(\mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*)^T\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}^*), \tag{6}$$

where $\mathbf{k}(\mathbf{x}^*) = [C(\mathbf{x}_1, \mathbf{x}^*), \ldots, C(\mathbf{x}_N, \mathbf{x}^*)]^T$ is the $N \times 1$ vector of covariances between the test and training cases, and $\kappa(x^*) = C(\mathbf{x}^*, \mathbf{x}^*)$ is the covariance between the test input itself.

The above modelling procedure was developed for modelling static nonlinearities, but it can be readily applied for modelling dynamic systems. A dynamic GP model is trained as the nonlinear autoregressive model with an exogenous input (NARX) representation, where the output at time step k depends on the delayed outputs y and the exogenous control inputs u:

$$\begin{aligned} y(k) = f(&y(k-1), \ldots, y(k-n), \\ &u(k-1), \ldots, u(k-m)) + \epsilon(k) \end{aligned} \tag{7}$$

where f denotes a function, $\epsilon(k)$ is white noise and the output $y(k)$ depends on the state vector $\mathbf{x}(k) = [y(k-1), y(k-2), \ldots, y(k-n), u(k-1), u(k-2), \ldots, u(k-m)]^T$ at time step k.

For the validation of obtained dynamic GP model the nonlinear output-error (NOE), also called parallel, model is used. This means that the NARX model is used to predict a further step ahead by replacing the data at instant k with the data at instant $k+1$ and using the prediction $\hat{y}(k)$ from the previous prediction step instead of the measured $y(k)$. This is then repeated indefinitely. The latter possibility is equivalent to simulation. Simulation, therefore, means that only on the basis of previous samples of a process input signal $u(k-i)$ can the model simulate future outputs. Frequently, the mean value of prediction $\hat{y}(k)$ is used to replace $y(k)$, which is called 'naive' simulation. Other possibilities, where the entire distribution is used, are described in, e.g., [2].

3 Advances in computation of dynamic GP models

3.1 On-line data selection and modelling

A noticeable drawback of system identification with GP models is the computation time necessary for the modelling. Regression based on GP models 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,e.g., [3, 4] as well as *on-line* modelling, e.g., [5], [6], 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.

A recent method for an *on-line* adapting of GP models is named Evolving Gaussian process models [7],[8]. Evolving systems [9] are self-developing systems inspired by the idea of system model evolution in a dynamically changing and evolving environment.

The GP models depend on data and covariance function. More detailed, data is defined with various regressors and basis vectors, and covariance function is defined with the type and hyperparameter values. Therefore, there are four parts that can evolve: regressors, basis vectors, type of covariance function and hyperparameter values. To overcome the computational greed necessary for GP model training only the subset of most informative data, so called basis vectors set, is used. With a type or a combination of various types of covariance function a prior knowledge of the system is included in the model. In dynamic non-linear systems, where the non-linear mapping between input and output can not be easily formulated, frequently the squared exponential covariance function is used presuming smoothness and stationarity of the system. That means the covariance function is fixed and does not need to evolve. With optimisation of hyperparameter values, uninfluential regressors have smaller values and as a consequence have smaller influence to the result. Therefore, all available regressors can be used. Consequently, only basis vectors set and hyperparameter values have left to be evolved. A general concept of evolving GP models, presented in [7] contains following steps: add new data to the set of most informative data, calculate information gain for all most informative data, remove worst data, calculate hyperparameter values and update covariance matrix. These basic steps are repeated for every in-coming sample of data until there is no more available data or until a requirement to stop the process is received.

3.2 Stochastic optimisation for initial values computation

The quality of GP models heavily depends on the covariance matrix. To calculate it, a covariance function needs to be selected according to the user's prior knowledge. Then the model can be further adjusted to data with appropriate tuning of hyperparameters. This can be done with various optimisation algorithms. For this purpose a conjugate gradient method is often used. Due to its deterministic nature, its success heavily depends on the initial values of hyperparameters, specially for complex systems. In this case stochastic methods seem appropriate. Three stochastic optimisation methods, i.e., Genetic Algorithm, Differential Evolution and Particle Swarm Optimisation, from the domain of evolutionary computation for hyperparameters optimisation were tested on a case study in [10]. The results from experimental work indicate that selected evolutionary algorithms, specially Differential Evolution and Particle Swarm Optimisation, successfully avoid local optima and find near-optimal values. Therefore, they seem useful, not so much for optimising multi-dimensional GP models, as for finding good initial values of hyperparameters.

3.3 Parallel processing for computing acceleration

The alternative to fight the computational limitation issues of GP modelling is to approach the computation problem from the utilised hardware technology point of view. Since hardware capabilities are increasing from day to day, this approach might seem inefficient when looking over the longer term, but it is undoubtedly effective in the short term. The demonstration and comparisson of the computational efficiency of Graphical Process Units for GP model identification that are affordable for most research groups via personal computer graphic cards is the topic of the paper [11].

The assessment on the dynamic-system identification and simulation case study revealed that using a Graphical Process Units computing architecture has benefits in accelerating the GP model identification and the MCMC simulation. It also demonstrated that the traditional naive approach to the simulation of a GP model does not benefit from GPU acceleration as a result of its sequential nature. As hardware capabilities are improving constantly and research on efficient algorithms is on-going, the presented assessment might not be of longterm value. However, it offers a state-of-the-art comparison of affordable hardware configuration that might help to circumvent the computational issue in an intermediate time frame before more efficient algorithms or better technology arrive.

4 Recent applications of dynamic GP models

4.1 Control systems design

The control design faces the challenge of more and more complex systems to be controlled. On the other hand the development of control technology has given the ability to design control for increasingly uncertain systems, es- pecially away from typical engineering fields, e.g., biosys- tems. The control based on GP models is an approach that may be promising for treating such systems. A number of publications where GP models have been used for the control design or as the part of controller appeared in recent years. A more thorough overview is given in [12]. These methods can be roughly divided into following groups:

- *Control based on models of inverse process dynamics;* Beside direct inverse model [13], which is of more theoretical value, Inverse Dynamics Control [14] and GP inverse model as feedforward [15],[14] have found their applications in robotics.

- *Model-based predictive control;* There are many different model-based predictive control structures that have been tested with GP models. These are: Internal Model Control [16], Predictive Functional Control [17], predictive control with general cost function [18] and approximate explicit stochastic nonlinear predictive control [19]. Only some of the references are cited for this very popular control-method family.

- *Gain-scheduling control;* A Fixed-Structure GP model is introduced in [20]. This kind of model is a model with predetermined linear structure with varying and probabilistic parameters represented by GP models. FSGP can be used for the gain-scheduling control design. In this case local controllers are designed for selected local model of the process.

- *Adaptive control;* Adaptive controller is the controller that continuously adapts to some changing process. When using the GP model for the adaptive control, different from gain-scheduling control, the GP model is identified on-line and this model is used in the control algorithm. Some of examples of such control algorithms are published in [7], [21], [22] and [23].

4.2 Fault detection

A fault detection approach based on GP model is proposed in papers [24] and [25]. The problem described is how to deal with insufficiently validated models during surveillance of nonlinear plants given the fact that tentative model-plant miss-match in such a case can cause false alarms. To avoid the risk, a novel model validity index was suggested in order to quantify the level of confidence associated to the detection results. This index is based on estimated distance between the current process data from data employed in the learning set.

4.3 Bayesian filtering

GP-BayesFilters [26] are a general framework for integrating GP prediction and observation models into Bayesian filtering techniques, including particle filters and extended and unscented Kalman filters. GP-BayesFilters have been shown to be extremely well suited for systems for which accurate parametric models are difficult to obtain. GP-BayesFilters learn nonparametric models from training data containing sequences of control inputs, observations, and ground truth states. Recent advances in this are can be found in papers, e.g., [27],[28],[26].

5 Trends, challenges and research opportunities

Several research topics remain not enough explored before GP models will become mature technology ready to use in the engineering practice.

Research opportunities can be roughly divided as follows: firstly to issues concerning dynamic systems modelling with GP models, secondly issues concerning the control design based on dynamic GP models and thirdly some general issues related to the GP modelling. The given list of issues is subjective and heavily based on the available information about on-going research activities throughout the world.

Modelling trends and issues. The computational burden that increases with increasing number of the data contained in the model, caused mainly by calculation of the inverse covariance matrix, directs researchers to find more efficient methods for the inverse covariance matrix calculation or the input data selection. The issue of automatically polishing data and finding informative portions is reported as one of key issues in dynamic system identification in general [29] and remains one of current challenges also in the GP modelling research.

The issue of recursive model identification is the issue that is closely linked to applications like adaptive control methods. In the machine learning community this kind of identification is known as on-line learning, which is not limited only to sequentially in-streaming data. The efficient method for recursive identification of the GP model still remains an unanswered challenge.

Further, methods for developing GP state-space models [30] still offer a lot of unsolved research problems.

Control design trends and issues. Most of the reported control methods have not addressed disturbance rejection, which is crucial in control systems, but have been more focused on the set-point tracking. The complete assessment of control methods requires also the disturbance rejection analysis, which in many cases still remain an unexplored issue.

First results that have a potential for control applications is also modelling of switching systems, e.g., [31].

If control methods are meant to be used in engineering practice more results on robust control design methodologies are necessary. GP models offer a lot of potential for the robust control design and offer a lot of research and application opportunities.

Some general issues that need to be brought-up are benchmarking of modelling as well as benchmarking of GP models' applications like control design methods, with purpose to assess different methods properly and fairly. A giant step to bring research results closer to the engineering practice is the integration of knowledge and software upgrade from pieces scattered around, mainly on the internet, into the user-friendly integrated software.

Research opportunities lay also in developing methods and procedures for various kinds analyses of dynamic-system models and closed-loop systems. Model-simulations stability and closed-loop stability are only two, very different, but important and challenging problems among many to be addressed.

6 Conclusions

The ever-increasing demands of new and existing applications are the driving factor behind the continued research into alternative system identification methods. Further the development of technology triggered the interest in modelling increasingly uncertain systems, especially away from typical engineering fields, e.g., biosystems. The design based on GP models is an approach that may be promising for treating such systems.

This paper provides the overview of some recent applications of GP models used for dynamic system identification described in various publications. Further, it lists some recent applications of dynamic GP models. These applications are control systems design, fault detection and Bayesian filtering.

The survey touched also possible trends, challenges and research opportunities. It was shown that a number of challenges and research opportunities that will bring the method to the level of maturity appropriate for engineering practice still lie ahead.

Acknowledgements: This work was supported by grants P2-0001 and J2-2099 of the Slovenian Research Agency.

References:

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

[2] J. Kocijan and B. Likar, "Gasliquid separator modelling and simulation with Gaussian-process models.," *Simulation Modelling Practice and Theory*, vol. 16, no. 8, pp. 910–922, 2008.

[3] J. Quinonero-Candela and C. E. Rasmussen, "A Unifying View of Sparse Approximate Gaussian Process Regression," *J. Mach. Learn. Res.*, vol. 6, pp. 1939–1959, 2005.

[4] J. Quinonero-Candela, C. E. Rasmussen, and C. K. I. Williams, "Approximation Methods for Gaussian Process Regression," tech. rep., Microsoft Research, September 2007.

[5] L. Csató and M. Opper, "Sparse online Gaussian processes," *Neural computation*, vol. 14, no. 3, pp. 641–668, 2002.

[6] D. Nguyen-Tuong and J. Peters, "Incremental online sparsification for model learning in real-time robot control," *Neurocomputing*, vol. 74, no. 11, pp. 1859–1867, 2011.

[7] D. Petelin and J. Kocijan, "Control system with evolving Gaussian process model," in *Proceedings of IEEE Symposium Series on Computational Intelligence, SSCI 2011*, (Paris), IEEE, 2011.

[8] D. Petelin, J. Kocijan, and A. Grancharova, "Online Gaussian process model for the prediciton of the ozone concentration in the air," *Comptes rendus de lAcadémie bulgare des Sciences*, vol. 64, no. 1, pp. 117–124, 2011.

[9] P. Angelov, D. P. Filev, and N. Kasabov, *Evolving Intelligent Systems: Methodology and Ap-*

plications. IEEE Press Series on Computational Intelligence, Wiley-IEEE Press, April 2010.

[10] D. Petelin, B. Filipič, and J. Kocijan, "Optimization of Gaussian process models with evolutionary algorithms," in *Proceedings of the 10th international conference on Adaptive and natural computing algorithms - Volume Part I*, ICANNGA'11, pp. 420–429, Springer-Verlag, 2011.

[11] B. Musizza, D. Petelin, and J. Kocijan, "Accelerated learning of gaussian process models," in *Proceedings of the 7th EUROSIM Congress on Modelling and Simulation EUROSIM 2010* (M. Šnorek, ed.), vol. 2, (Prague), p. 8, 2010.

[12] J. s Kocijan, "Control Algorithms Based on Gaussian Process Models: A State-of-the-Art Survey," in *Proceedings Volume from the Special International Conference on Complex Systems: Synergy of Control, Communications and Computing - COSY 2011*, (Ohrid, Republic of Macedonia), pp. 69–80, 2011.

[13] Y. Engel, P. Szabo, and D. Volkinshtein, "Learning to Control an Octopus Arm with Gaussian Process Temporal Difference Methods," in *Advances in Neural Information Processing Systems* (Y. Weiss, B. Schoelkopf, and J. Platt, eds.), vol. 18, pp. 347–354, MIT Press, 2006.

[14] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Computed torque control with nonparametric regression models," in *Proceedings of the 2008 American Control Conference, ACC 2008*, (Seattle, Washington), p. 6 pages, 2008.

[15] D. Nguyen-Tuong, J. Peters, M. Seeger, and B. Schoelkopf, "Learning inverse dynamics: a comparison," in *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, (Bruges, Belgium), pp. 13–18, 2008.

[16] G. Gregorčič and G. Lightbody, "Gaussian process internal model control," *International Journal of Systems Science*, pp. 1–16, 2011.

[17] J. Kocijan and R. Murray-Smith, "Nonlinear predictive control with Gaussian process model," in *Switching and Learning in Feedback Systems* (R. Murray-Smith and R. Shorten, eds.), vol. 3355 of *Lecture Notes in Computer Science*, pp. 185–200, Springer, 2005.

[18] A. Grancharova and J. Kocijan, "Stochastic predictive control of a thermoelectric power plant," in *Proceedings of the International Conference Automatics and Informatics07*, (Sofia), pp. I–13–I–16, 2007.

[19] A. Grancharova, J. Kocijan, and T. A. Johansen, "Explicit stochastic predictive control of combustion plants based on gaussian process models," *Automatica*, vol. 44, no. 4, pp. 1621–1631, 2008.

[20] K. Ažman and J. Kocijan, "Fixed-structure Gaussian process model," *International Journal of Systems Science*, vol. 40, no. 12, pp. 1253–1262, 2009.

[21] D. Sbarbaro and R. Murray-Smith, "An adaptive nonparametric controller for a class of nonminimum phase non-linear system," in *Proceedings of IFAC 16th World Congress*, (Prague, Czech Republic), 2005.

[22] C. E. R. Marc Peter Deisenroth and J. Peters, "Gaussian process dynamic programming," *Neurocomputing*, vol. 72, no. 7–9, pp. 1508–1524, 2009.

[23] D. Nguyen-Tuong, M. Seeger, and J. Peters, *Real-time local GP model learning*, vol. 264, ch. From Motor Learning to Interaction Learning in Robots, pp. 193–207. Springer, 2010.

[24] D. Jurčić and J. Kocijan, "Fault detection based on Gaussian process model," in *Proceedings of the 5th Vienna Symposium on Mathematical Modeling (MathMod)* (I. Troch and F. Breitenecker, eds.), (Vienna, Austria), 2006.

[25] D. Juričić, P. Ettler, and J. Kocijan, "Fault detection based on Gaussian process models: An Application to the Rolling Mill," in *Proceedings of the 8th International Conference on Informatics in Control, Automation and Robotics - ICINCO 2011*, vol. 1, (Noordwijkerhout, The Netherlands), pp. 437–440, 2011.

[26] J. Ko and D. Fox, "Learning GP-BayesFilters via Gaussian process latent variable models," *Autonomous Robots*, vol. 30, pp. 3–23, 2011.

[27] M. Deisenroth, M. Huber, and U. Hannebeck, "Analytic moment-based Gaussian process filtering," in *Proceedings of the 26th Annual International Conference on Machine Learning*, (Montreal, Canada), pp. 225–232, 2009.

[28] S. Reece and S. J. Roberts, "An introduction to Gaussian processes for the Kalman filter expert," in *Proceedings of the 13th International Conference on Information Fusion*, (Edinburgh), 2010.

[29] L. Ljung, "Perspectives on system identification," in *Proceedings of IFAC 17th World Congress 2008*, (Seoul), p. 6 pages, 2008.

[30] R. Turner, M. P. Deisenroth, and C. E. Rasmussen, "State-space inference and learning with Gaussian processes," in *Proceedings of 13th International Conference on Artificial Intelligence and Statistics*, vol. 9, (Sardinia, Italy), pp. 868–875, 2010.

[31] Y. Saatçi, R. Turner, and C. E. Rasmussen, "Gaussian Process change point models," in *Proceedings of the 27th Annual International Conference on Machine Learning*, pp. 927–934, 2010.