SCIENCE CHINA
Information Sciences

• LETTER •

December 2018, Vol. 61 129206:1–129206:3
https://doi.org/10.1007/s11432-018-9634-7

Bayesian random Fourier filters for Gaussian noises

Shiyuan WANG[1,2*], Wanli WANG[1,2,3], Shukai DUAN[1,2],
Lidan WANG[1,2] & Chi K. TSE[3]

[1]College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China;
[2]Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, Chongqing 400715, China;
[3]Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong, China

Received 6 July 2018/Revised 6 August 2018/Accepted 30 September 2018/Published online 22 November 2018

Citation Wang S Y, Wang W L, Duan S K, et al. Bayesian random Fourier filters for Gaussian noises. Sci China Inf Sci, 2018, 61(12): 129206, https://doi.org/10.1007/s11432-018-9634-7

Dear editor,

Signal processing algorithms are generally used to identify an unknown system, and can be classified as batch algorithms and online learning algorithms (OLAs). Batch algorithms are applied to offline scenarios that require all data being available. In real-time applications, because information is processed sequentially, an OLA is desirable [1].

As a commonly used OLA, the kernel adaptive filter (KAF) [2] is proposed to solve complicated nonlinear issues. However, the network size of KAF is linearly growing, thus leading to high computational overhead. To this end, a random feature mapping function (RFMF) is used to generate the online sequential extreme learning machine (OS-ELM) with a fixed network size in the random Fourier feature space (RFFS) [3]. However, the statistical characteristics of system parameters are not considered in OS-ELM. Bayesian inference provides a probability distribution on a system model and its parameters, and is applied to learning algorithms for improving generalization performance [4].

In this study, a novel OLA with a fixed network size, namely online Bayesian random Fourier filter (BRFF), is proposed. The main contributions are summarized as follows: (1) The network size of online BRFF is fixed; (2) In comparison with KAF, online BRFF can improve the filtering accuracy for Gaussian noises by applying Bayesian inference and a Gaussian diffusion process to estimate the system parameters in the RFFS; (3) A Lyapunov function in the RFFS is established to validate the convergence of online BRFF.

Problem formulation. In a nonlinear system, a sequence of input-output pairs is assumed to be denoted by $\{(\boldsymbol{x}_k, y_k), k = 1, 2, \ldots \mid \boldsymbol{x}_k \in \mathbb{R}^d, y_k \in \mathbb{R}\}$. The learned input-output relationship existing in this system is therefore described by

$$y_k = f(\boldsymbol{x}_k) + v_k, \tag{1}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ and v_k is a Gaussian noise with zero mean and variance δ_n^2. The purpose of an OLA is to obtain an approximation $\hat{f}$ of f from the available noisy input-output data.

Online Bayesian random Fourier filter. A random mapping $\boldsymbol{z}(\boldsymbol{x})$ with randomized Fourier features [5] projects the input data $\boldsymbol{x}$ of dimension d onto another Euclidean feature space of a relatively high dimension D ($D > d$) using randomly generated data, i.e.,

$$\boldsymbol{z}(\boldsymbol{x}) = \sqrt{2/D}\big[\cos(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\omega}_1 + b_1), \cos(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\omega}_2 + b_2), \ldots, \cos(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\omega}_D + b_D)\big]^{\mathrm{T}}, \tag{2}$$

where weight $\boldsymbol{\omega} \in \mathbb{R}^d$ is generated from a Gaussian distribution denoted by $\mathcal{N}(\boldsymbol{\omega}; \boldsymbol{0}, \sigma^2\boldsymbol{I})$ and bias b is obtained from a uniform distribution in $[0, 2\pi]$. Based on $\boldsymbol{z}(\boldsymbol{x}_k) \in \mathbb{R}^D$, the function approximation $\hat{f}$ takes the form of the linear inner product in the RFFS, i.e.,

* Corresponding author (email: wsy@swu.edu.cn)

© Science China Press and Springer-Verlag GmbH Germany, part of Springer Nature 2018

$$\hat{f}(\boldsymbol{x}_k) = \boldsymbol{z}(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{\theta}_k, \tag{3}$$

where $\boldsymbol{\theta}_k$ denotes the weight in the RFFS. In the following, Bayesian inference based on the assumption of Gaussian distribution for the weight and noise is used to estimate $\boldsymbol{\theta}_k$ in the network with the fixed dimension D shown in (2).

First, a Gaussian diffusion process associated with a Gaussian distribution noise $\mathcal{N}(\boldsymbol{q}; \boldsymbol{0}, \delta_D^2\boldsymbol{I})$ is used to model the system parameters, i.e.,

$$p(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1}) = \mathcal{N}(\boldsymbol{\theta}_k; \boldsymbol{\theta}_{k-1}, \delta_D^2\boldsymbol{I}), \tag{4}$$

where $\boldsymbol{I}$ denotes the identity matrix and δ_D^2 is the variance of the diffusion noise in each dimension.

Then, the output $y_k \in \mathbb{R}$ is described by the following probabilistic model:

$$p(y_k|\boldsymbol{\theta}_k) = \mathcal{N}(y_k; \boldsymbol{z}(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{\theta}_k, \delta_n^2), \tag{5}$$

where δ_n^2 is the variance of the observation noise.

Finally, the approximation $\hat{f}$ in the RFFS can be described by (4) and (5) from the perspective of probabilistic distribution.

The posterior density of weights at discrete time $k-1$ is denoted by $p(\boldsymbol{\theta}_{k-1}|y_{1:k-1}) = \mathcal{N}(\boldsymbol{\theta}_{k-1}; \hat{\boldsymbol{\theta}}_{k-1}, \boldsymbol{P}_{k-1})$. The online BRFF calculates the posterior density $p(\boldsymbol{\theta}_k|y_{1:k})$ at discrete time k upon receipt of a new measurement y_k, which consists of a prediction step and an update step.

• Prediction step. The predictive density of weights at discrete time k is calculated using the Chapman-Kolmogorov equation [6], i.e.,

$$p(\boldsymbol{\theta}_k|y_{1:k-1}) = \int p(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1})p(\boldsymbol{\theta}_{k-1}|y_{1:k-1})\mathrm{d}\boldsymbol{\theta}_{k-1}, \tag{6}$$

where $\hat{\boldsymbol{\theta}}_{k|k-1}$ and $\boldsymbol{P}_{k|k-1}$ in $p(\boldsymbol{\theta}_k|y_{1:k-1}) = \mathcal{N}(\boldsymbol{\theta}_k; \hat{\boldsymbol{\theta}}_{k|k-1}, \boldsymbol{P}_{k|k-1})$ are denoted by

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{P}_{k-1} + \delta_D^2\boldsymbol{I}, \tag{7}$$

$$\hat{\boldsymbol{\theta}}_{k|k-1} = \hat{\boldsymbol{\theta}}_{k-1}. \tag{8}$$

• Update step. The posterior density of weights at discrete time k is computed by Bayes' rule, i.e.,

$$p(\boldsymbol{\theta}_k|y_{1:k}) = \frac{p(y_k|\boldsymbol{\theta}_k)p(\boldsymbol{\theta}_k|y_{1:k-1})}{\int p(y_k|\boldsymbol{\theta}_k)p(\boldsymbol{\theta}_k|y_{1:k-1})\mathrm{d}\boldsymbol{\theta}_k}, \tag{9}$$

where the current observation y_k given the current weight $\boldsymbol{\theta}_k$ is assumed to be conditionally independent of the observation histories, i.e., $p(y_k|\boldsymbol{\theta}_k, y_{1:k-1}) = p(y_k|\boldsymbol{\theta}_k)$.

According to (5) and (6), the joint distribution of $\boldsymbol{\theta}_k$ and y_k given $y_{1:k-1}$ is calculated by

$$p(\boldsymbol{\theta}_k, y_k|y_{1:k-1}) = p(\boldsymbol{\theta}_k|y_{1:k-1})p(y_k|\boldsymbol{\theta}_k). \tag{10}$$

According to Lemma 1 in Appendix A, by setting $\boldsymbol{x} = \boldsymbol{\theta}_k$, $\boldsymbol{y} = y_k$, $\boldsymbol{H} = \boldsymbol{z}(\boldsymbol{x}_k)^{\mathrm{T}}$, $\boldsymbol{m} = \hat{\boldsymbol{\theta}}_{k|k-1}$, $\boldsymbol{Q}_1 = \boldsymbol{P}_{k|k-1}$, and $\boldsymbol{Q}_2 = \delta_n^2$ [6], we obtain the mean and covariance of $p(\boldsymbol{\theta}_k, y_k|y_{1:k-1})$ as follows:

$$\boldsymbol{\theta}_k^y = \begin{bmatrix} \hat{\boldsymbol{\theta}}_{k|k-1} \\ y_k^- \end{bmatrix}, \ \boldsymbol{P}_k^{\boldsymbol{\theta},y} = \begin{bmatrix} \boldsymbol{P}_{k|k-1} & \boldsymbol{P}_{k|k-1}^z \\ \boldsymbol{P}_{k|k-1}^{z\mathrm{T}} & \gamma_n^2 \end{bmatrix}, \tag{11}$$

where $y_k^- = \boldsymbol{z}(\boldsymbol{x}_k)^{\mathrm{T}}\hat{\boldsymbol{\theta}}_{k|k-1}$ and $\gamma_n^2 = \boldsymbol{z}(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{P}_{k|k-1} \cdot \boldsymbol{z}(\boldsymbol{x}_k) + \delta_n^2$, $\boldsymbol{P}_{k|k-1}^z = \boldsymbol{P}_{k|k-1}\boldsymbol{z}(\boldsymbol{x}_k)$.

Based on the joint distribution in (11), the conditional distribution of $\boldsymbol{\theta}_k$ given $y_{1:k}$ can be obtained by $p(\boldsymbol{\theta}_k|y_{1:k}) = \mathcal{N}(\boldsymbol{\theta}_k; \hat{\boldsymbol{\theta}}_k, \boldsymbol{P}_k)$ with

$$\boldsymbol{P}_k = \boldsymbol{P}_{k|k-1} - \frac{\boldsymbol{P}_{k|k-1}\boldsymbol{z}(\boldsymbol{x}_k)\boldsymbol{z}(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{P}_{k|k-1}}{\boldsymbol{z}(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{P}_{k|k-1}\boldsymbol{z}(\boldsymbol{x}_k) + \delta_n^2}, \tag{12}$$

$$\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \boldsymbol{G}_k e_k, \tag{13}$$

where $e_k = y_k - \boldsymbol{z}(\boldsymbol{x}_k)^{\mathrm{T}}\hat{\boldsymbol{\theta}}_{k-1}$ and the gain matrix $\boldsymbol{G}_k$ takes the form of

$$\boldsymbol{G}_k = \frac{\boldsymbol{P}_{k|k-1}\boldsymbol{z}(\boldsymbol{x}_k)}{\boldsymbol{z}(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{P}_{k|k-1}\boldsymbol{z}(\boldsymbol{x}_k) + \delta_n^2}. \tag{14}$$

The proposed online BRFF is summarized in Algorithm 1.

Algorithm 1 Online Bayesian random Fourier filter

Initiation: δ_n^2, δ_D^2, D, $\boldsymbol{P}_1$, and $\hat{\boldsymbol{\theta}}_1$.
while $\{\boldsymbol{x}_k, y_k\}$ $(k > 1)$ available **do**
(1) Transform the input data by (2);
(2) Calculate $\boldsymbol{P}_{k|k-1}$ and $\boldsymbol{G}_k$ by (7) and (14);
(3) Update $\boldsymbol{P}_k$ and $\hat{\boldsymbol{\theta}}_k$ and by (12) and (13);
end whlile

Remark 1. Because the online BRFF is obtained from Bayesian inference, it has a similar update form shown in Algorithm 1 to that of nonlinear Kalman filter (NKF) [7,8]. However, in comparison with NKF estimating the unknown state vector from observations, BRFF can learn the nonlinear function hidden in input-output data. In addition, NKFs perform state estimation in the input space whereas BRFF performs the function approximation in a D-dimensional RFFS.

Batch BRFF. When all the input-output pairs are available for training, the weights in the RFFS can also be determined by the batch BRFF based on Gaussian process regression (GPR) [6]. Define L transformed input data and observations as $\boldsymbol{Z}_L = [\boldsymbol{z}(\boldsymbol{x}_1), \ldots, \boldsymbol{z}(\boldsymbol{x}_L)]$ and $\boldsymbol{y}_L = [y_1, \ldots, y_L]^{\mathrm{T}}$, respectively. Under the assumption of conditional independence of observations, the full likelihood of the observations given the weight vector is

$$p(y_{1:L}|\boldsymbol{\theta}) = \prod_{i=1}^{L} p(y_i|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}_L; \boldsymbol{Z}_L^{\mathrm{T}}\boldsymbol{\theta}, \delta_n^2\boldsymbol{I}). \tag{15}$$

Assume that the weight vector has a prior distribution of $\mathcal{N}(\boldsymbol{\theta}; \mathbf{0}, \delta_\theta^2 \boldsymbol{I})$ where the mean value $\mathbf{0}$ is the null vector with appropriate dimension.

Let the posterior distribution of the weight vector given $y_{1:L}$ be $p(\boldsymbol{\theta}|y_{1:L}) = \mathcal{N}(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}_L, \boldsymbol{P}_L)$. According to Lemma 1 in Appendix A, by letting $\boldsymbol{x} = \boldsymbol{\theta}_k$, $\boldsymbol{y} = \boldsymbol{y}_L$, $\boldsymbol{H} = \boldsymbol{Z}_L^{\mathrm{T}}$, $\boldsymbol{m} = \mathbf{0}$, $\boldsymbol{Q}_1 = \delta_\theta^2 \boldsymbol{I}$, and $\boldsymbol{Q}_2 = \delta_n^2 \boldsymbol{I}$, we obtain the mean and covariance of $p(\boldsymbol{\theta}|y_{1:L})$ as follows:

$$\hat{\boldsymbol{\theta}}_L = (\boldsymbol{Z}_L \boldsymbol{Z}_L^{\mathrm{T}} + \delta_n^2 \delta_\theta^{-2} \boldsymbol{I})^{-1} \boldsymbol{Z}_L \boldsymbol{y}_L, \tag{16}$$

$$\boldsymbol{P}_L = (\delta_n^{-2} \boldsymbol{Z}_L \boldsymbol{Z}_L^{\mathrm{T}} + \delta_\theta^{-2} \boldsymbol{I})^{-1}. \tag{17}$$

The batch BRFF with $\mathcal{O}(L^3)$ incurs a higher computational cost than the online BRFF with $\mathcal{O}(D^2)$ at each iteration. Here, the batch BRFF is only used for performance comparison.

Stability of online BRFF. A Lyapunov function [9] is constructed to perform the stability analysis of online BRFF.

Define the predicted error and the estimated error as $\tilde{\boldsymbol{\theta}}_{k|k-1} = \boldsymbol{\theta}_k - \hat{\boldsymbol{\theta}}_{k|k-1}$ and $\tilde{\boldsymbol{\theta}}_k = \boldsymbol{\theta}_k - \hat{\boldsymbol{\theta}}_k$, respectively. According to (4) and (8), the following equality can be established:

$$\tilde{\boldsymbol{\theta}}_{k|k-1} = \boldsymbol{\beta}_{k-1} \tilde{\boldsymbol{\theta}}_{k-1}, \tag{18}$$

where the unknown diagonal matrix $\boldsymbol{\beta}_k = \mathrm{diag}([\boldsymbol{\beta}_{k,1}, \ldots, \boldsymbol{\beta}_{k,D}])$ includes the perturbations with consideration of the diffusion noise in (4). Similarly, we have e_k in (13) re-arranged as

$$\alpha_k e_k = \boldsymbol{z}(\boldsymbol{x}_k)^{\mathrm{T}} \tilde{\boldsymbol{\theta}}_{k|k-1}, \tag{19}$$

where α_k is an unknown scalar with consideration of the observation noise in (5).

To perform stability analysis of online BRFF, we construct the Lyapunov function by $V_k = \tilde{\boldsymbol{\theta}}_k^{\mathrm{T}} \boldsymbol{P}_k^{-1} \tilde{\boldsymbol{\theta}}_k$, which is a quadratic function associated with the estimated error $\tilde{\boldsymbol{\theta}}_k$ in the RFFS. The stability of online BRFF is gualranteed by the decreasing characteristic of V_k given in Theorem 1.

Theorem 1. The Lyapunov function at discrete time k, $V_k \leqslant V_{k-1}$ is satisfied, if the following inequalities hold:

(1) There exist $p_{\min}$, $p_{\max}> 0$ and $p_{\max} > p_{\min}$ such that

$$p_{\min} \boldsymbol{I} \leqslant \boldsymbol{P}_{k-1} \leqslant p_{\max} \boldsymbol{I}. \tag{20}$$

(2) $\boldsymbol{\beta}_{k,j}$ with $j = 1, \ldots, D$ in (18) is bounded as follows:

$$|\boldsymbol{\beta}_{k,j}| \leqslant 1. \tag{21}$$

(3) α_k in (19) satisfies

$$1 - \sqrt{1 - \gamma_k} \leqslant \alpha_k \leqslant 1 + \sqrt{1 - \gamma_k}, \tag{22}$$

where γ_k is given by

$$\gamma_k = \frac{\boldsymbol{z}(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{P}_{k|k-1} \boldsymbol{z}\,(\boldsymbol{x}_k)}{\boldsymbol{z}(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{P}_{k|k-1} \boldsymbol{z}\,(\boldsymbol{x}_k) + \delta_n^2}. \tag{23}$$

The proof and the simulated results are given in Appendixes B–D.

Conclusion and future work. A novel online BRFF for Gaussian noise is proposed by using random Fourier mapping and Bayesian theory. The accuracy and trackability of online BRFF is significantly improved. Because the online BRFF has a fixed network structure, the computational complexity is reduced. A batch BRFF is also presented for comparison. In addition, the convergence analysis of online BRFF is derived by using the constructed Lyapunov function to verify its stability. In the future work, the dimension of the random Fourier feature space will be discussed theoretically.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61671389, 61672436).

Supporting information Appendixes A–D. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Kivinen J, Smola A J, Williamson R C. Online learning with kernels. IEEE Trans Signal Process, 2004, 52: 2165–2176
2 Ma W T, Duan J D, Man W S, et al. Robust kernel adaptive filters based on mean p-power error for noisy chaotic time series prediction. Eng Appl Artif Intel, 2017, 58: 101–110
3 Deng C W, Huang G B, Xu J, et al. Extreme learning machines: new trends and applications. Sci China Inf Sci, 2015, 58: 020301
4 Rasmussen C E, Williams C K I. Gaussian Processes for Machine Learning. Cambridge: MIT Press, 2006
5 Bouboulis P, Pougkakiotis S, Theodoridis S. Efficient KLMS and KRLS algorithms: a random Fourier feature perspective. In: Proceedings of IEEE Workshop on Statistical Signal Processing, Palma de Mallorca, 2016. 1–5
6 Särkkä S. Bayesian Filtering and Smoothing. Cambridge: Cambridge University Press, 2013
7 Zeng N Y, Wang Z D, Zhang H. Inferring nonlinear lateral flow immunoassay state-space models via an unscented Kalman filter. Sci China Inf Sci, 2016, 59: 112204
8 Zhang Y G, Huang Y L. Gaussian approximate filter for stochastic dynamic systems with randomly delayed measurements and colored measurement noises. Sci China Inf Sci, 2016, 59: 092207
9 Zhao C, Guo L. PID controller design for second order nonlinear uncertain systems. Sci China Inf Sci, 2017, 60: 022201