Ensemble Kalman Filtering for Online Gaussian Process Regression and Learning

Danil Kuzin[*], Le Yang[†], Olga Isupova[‡] and Lyudmila Mihaylova[§]

[*]Department of Automatic Control and Systems Engineering
The University of Sheffield, Sheffield, UK
Email: dkuzin1@sheffield.ac.uk

[†] Department of Electrical and Computer Engineering
University of Canterbury, Christchurch, New Zealand
E-mail: le.yang.le@gmail.com

[‡]Department of Engineering Science
The University of Oxford, Oxford, UK
Email: olga.isupova@eng.ox.ac.uk

[§]Department of Automatic Control and Systems Engineering
The University of Sheffield, Sheffield, UK
Email: l.s.mihaylova@sheffield.ac.uk

Abstract—Gaussian processes are used in Bayesian machine learning and signal processing for estimation of unknown functions. However, they suffer from high computational complexity, as in a basic form they scale cubically with the number of observations. Several approaches based on inducing points were proposed to handle this problem in a static context. However, these methods lack performance for online data that is received sequentially over time. In this paper, a novel online algorithm for training sparse Gaussian process models is presented. It treats the mean and hyperparameters of the Gaussian process as the state and parameters of the ensemble Kalman filter, respectively. The online evaluation of the parameters and the state is performed on new upcoming samples of data. This procedure iteratively improves the accuracy of parameter estimates. The ensemble Kalman filter reduces the computational complexity required to obtain predictions with Gaussian processes preserving the accuracy level of these predictions. The performance of the proposed method is demonstrated on synthetic dataset and real large dataset of UK house prices.

I. INTRODUCTION

In Bayesian machine learning and signal processing, Gaussian processes (GPs) are used to approximate unknown functions ([1]) and provide posterior estimates for mean and variance of the target functions in the selected points. The function can be latent, and, in this case, GPs represent the idea of proximity, or structure, when close values of inputs lead to close values of outputs. Another popular application is black-box optimisation with GPs, known as Bayesian optimisation. GPs are widely applied for signal processing, examples include audio ([2]), communications ([3]), fault detection ([4]).

GPs are characterised by covariance functions that usually have a set of hyperparameters. The popular examples are squared-exponential, Matérn and exponential covariance functions ([1]). They are stationary functions that depend only on distance between points, they provide solutions with different smoothness properties. The hyperparameters are hard to estimate by experts and they are usually learnt within the GP framework, for example by optimising the marginal likelihood, which leads to local maxima.

GPs are usually represented in a grid of points and it is the source of the main limitation. The required resources are huge: computational time scales cubically with the number of grid points, required memory scales quadratically. It is essential to reduce these numbers in order to make GPs applicable for larger datasets or online inference.

During the last decades multiple approaches have been proposed to deal with this problem. The most popular approach is introduction of inducing points [5] where the locations of grid points are optimised, their amount is reduced with an attempt to maintain good prediction power. In [6] inducing points are treated as variational parameters and Bayesian inference is performed. In [7] expectation propagation is proposed for the Bayesian inference of the parameters.

Another approach is the distributed computations, where local predictions are combined into unified mean and variance predictions. The idea of partitioning dataset for the GP problem was considered in [8] with use of Kd-trees. The distributed Bayesian version with sparse approximation is proposed in [9].

In [10] the online procedure for updating GP parameters is proposed. The mean in the grid points is treated as a state variable, GP hyperparameters and noise are treated as parameters and for the joint state-parameter vector the unscented Kalman filter is used. The model has been recently used for the received-signal-strength estimation ([11], [12]), flow modelling and prediction in sports analytics ([13]). Sampling approaches for the online updating of the GP hyperparameters include slice sampler ([14]), sequential Monte Carlo ([4]), Bayesian Monte Carlo ([15]).

In this paper the ensemble Kalman filter is proposed to deal with online GP problem. It provides more stable parameter estimates with better predictive performance.

The main contributions of this paper can be summarised as:

- For the first time the ensemble Kalman filter (EnKF) for the problem of online GP regression and learning is proposed. This allows to reduce the computational complexity related to the prediction, as the size of the invertible matrices is reduced according to the ensemble sizes.
- The dual and joint versions of the ensemble Kalman filter are presented in the paper.
- The performance of the algorithms is compared using the synthetic dataset and real large dataset of the house prices.

The paper is organised in the following way: first the overview of the ensemble Kalman filter and the problem of state and parameter estimation within this framework is described in Section II. In Section III the joint and dual EnKF frameworks for GPs are described. In Section IV the experiments are conducted on synthetic data and UK house price data and the conclusion is presented in Section V.

II. ENSEMBLE KALMAN FILTER OVERVIEW

Ensemble Kalman filter was originally discussed in [16], and a recent overview with different improvement techniques is given in [17]. EnKF uses the Monte Carlo method to generate an ensemble of state sigma points and then this state ensemble is passed through the measurement function to obtain the observation ensemble, it is additionally perturbed with the measurement noise. The mean and variance of the resulting observational distribution together with actual observations are used to update the state. The main computational difference in comparison to the classic Kalman filter is that the covariance matrices are replaced with ensembles that can be less in dimensionality.

The usual approach to parameters estimation is augmenting the state vector with parameters vector thus creating the larger augmented state-parameter vector. It can then be used to perform the online estimation within the EnKF framework [18], [19]. In [20] dual estimation of state and parameters is proposed to replace joint estimation as in classic Kalman filters: for every new observation, first the parameters are updated and then using the updated parameters the state is updated. Dual estimation of parameters and state for EnKF is considered in [21].

Other approaches for parameters estimation in EnKF include the maximum likelihood method [22], [23] and the Bayesian inference [24].

III. ENSEMBLE KALMAN FILTER FOR GAUSSIAN PROCESSES

This paper proposes the algorithm for the problem of online estimation of the constant unknown continuous function $f(\mathbf{x})$ of the D-dimensional input vector $\mathbf{x} \in \mathbb{R}^D$. The unknown function is approximated with a GP: the mean $\mathbf{g} \in \mathbb{R}^K$ of the GP is approximated at the K grid points $\mathbf{X}_g \in \mathbb{R}^{K \times D}$ and L_θ parameters of the covariance function $\boldsymbol{\theta} \in \mathbb{R}^{L_\theta}$ are estimated. With mean and parameters of the covariance function it is

possible to predict the mean and variance of $f(\mathbf{x}^*)$ at any point $\mathbf{x}^*$.

It is assumed that the observations of the function are available sequentially, at every timestamp $1 \le t \le T$, where T is the last observation timestamp. At every iteration t of the algorithm a total of S one-dimensional noisy function observations $\mathbf{y}_t \in \mathbb{R}^S$ are obtained at random points $\mathbf{x}_{new}$. The variance of noise σ_y^2 assumed to be unknown and it is estimated at every iteration of the algorithm. The full vector of parameters is therefore $\boldsymbol{\eta} = [\boldsymbol{\theta}, \sigma_y^2] \in \mathbb{R}^L$, where $L = L_\theta + 1$.

The dependency between covariance function parameters and observations is non-linear, therefore a nonlinear version of Kalman filter is required. The ensemble Kalman filter allows to have constant complexity for updates, which is determined by the number of ensemble points, N.

Two versions of ensemble Kalman filter for the online GP learning are proposed, they differ in the way how hyperparameters of the GP are treated: *Dual EnKF* first updates the hyperparameters of the GP and then based on their estimates updates the state; *Joint EnKF* updates hyperparameters of the GP and the state simultaneously with the augmented state–hyperparameter vector.

A. Dual Ensemble Kalman Filter for Gaussian Processes

This algorithm is further denoted as Dual GP-EnKF. It uses the ensembles of same size N, to approximate the distributions of parameters and state. At every iteration the predicted distributions of parameters and state are computed, and the observations are predicted. Then, based on the cross-covariance of parameters and observations ensembles, Kalman gain is computed and it is used to update parameters distribution. After this step, new observations are predicted with updated parameters and then cross-covariance of new observations and state is used to update state. The details of the algorithm are presented below.

1) Initialisation: Initially, ensembles for parameters $\mathbf{H} \in \mathbb{R}^{N \times L} = [\boldsymbol{\eta}^{(i)} \in \mathbb{R}^{1 \times L}]_{1 \le i \le N}$ and mean $\mathbf{G} \in \mathbb{R}^{N \times K} = [\mathbf{g}^{(i)} \in \mathbb{R}^{1 \times K}]_{1 \le i \le N}$ at the grid points of the GP are generated. The rows of matrices correspond to ensemble members. For parameters that can only be positive, such as variance, logarithms of their values are used in the ensemble. Initial ensembles are generated from the Gaussian distribution: for each ensemble index $1 \le i \le N$

$$\boldsymbol{\eta}_{0|0}^{(i)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_H), \tag{1a}$$

$$\mathbf{g}_{0|0}^{(i)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_G), \tag{1b}$$

where $\boldsymbol{\Sigma}_H$, $\boldsymbol{\Sigma}_G$ are the initial covariance matrices for the ensembles. In our experiments, they are assumed to be diagonal. $\mathcal{N}(\cdot)$ denotes the Gaussian distribution.

After the initialisation at every iteration of the algorithm three steps follow: prediction, update for parameters, update for state.

2) Prediction: For the whole running time of the algorithm the estimated function remains constant, while unknown. This

can be simulated with the random walk motion model for the parameters and state, each ensemble member is updated as

$$\boldsymbol{\eta}_{t+1|t}^{(i)} = \boldsymbol{\eta}_{t|t}^{(i)} + \boldsymbol{\varepsilon}_\eta, \tag{2a}$$

$$\mathbf{g}_{t+1|t}^{(i)} = \mathbf{g}_{t|t}^{(i)} + \boldsymbol{\varepsilon}_g, \tag{2b}$$

where $\boldsymbol{\varepsilon}_\eta \sim \mathcal{N}(\mathbf{0}, \sigma_\eta \mathbf{I})$, $\boldsymbol{\varepsilon}_g \sim \mathcal{N}(\mathbf{0}, \sigma_g \mathbf{I})$ are the noise variables with corresponding variances. We also consider Liu-West filter that has other approach to parameter prediction later in this section.

Assume that S observations were obtained at locations $\mathbf{X}_{\text{new}} = [\mathbf{x}_{\text{new}}^s]_{s=1}^S$. According to the definition of GPs the joint distribution for any discrete set of samples is the multivariate Gaussian distribution. Therefore, for each parameter ensemble i the distribution of predicted function values $\widehat{\mathbf{y}}^{(i)} = [\widehat{y}^{1,(i)}, \dots, \widehat{y}^{S,(i)}]$ at locations $\mathbf{x}_{\text{new}}$ can be obtained as

$$\widehat{\mathbf{y}}^{(i)} = K(\mathbf{X}_{\text{new}}, \mathbf{X}_g | \boldsymbol{\theta}_{t+1|t}^{(i)}) \times [K(\mathbf{X}_g, \mathbf{X}_g | \boldsymbol{\theta}_{t+1|t}^{(i)}) + \sigma_{y\,t+1|t}^{2(i)} \mathbf{I}]^{-1} \mathbf{g}_{t+1|t}^{(i)}, \tag{3}$$

where $K(\mathbf{X}_1, \mathbf{X}_2 | \boldsymbol{\theta})$ is the covariance matrix evaluated at every pair of points from $\mathbf{X}_1$, $\mathbf{X}_2$ with parameters $\boldsymbol{\theta}$; $\boldsymbol{\theta}_{t+1|t}^{(i)}$ and $\sigma_{y\,t+1|t}^{2(i)}$ are components of the joint parameter vector $\boldsymbol{\eta}_{t+1|t}^{(i)}$. The matrix for all predictions is denoted as $\widehat{\mathbf{Y}} \in \mathbb{R}^{N \times S} = [\widehat{\mathbf{y}}^{(i)}]_{i=1}^N$

In EnKF, observations are treated as random variables and the observation ensemble is generated, which has a Gaussian distribution around the actual observation with predefined covariance σ_{obs}^2

$$\mathbf{y}^{(i)} = \mathbf{y} + \boldsymbol{\varepsilon}_{\text{obs}}, \tag{4}$$

where $\boldsymbol{\varepsilon}_{\text{obs}} \sim \mathcal{N}(0, \sigma_{\text{obs}}^2 \mathbf{I})$

3) Update parameters: EnKF updates are similar to the usual KF, with the means and covariances estimated from the ensembles. First, cross covariances of parameter ensemble and prediction ensemble are computed. Let $\mathbb{E}_i[\cdot]$ denote the expected value with respect to ensembles. Then

$$\overline{\boldsymbol{\eta}}_{t+1|t} = \frac{1}{N} \sum_{i=1}^N \boldsymbol{\eta}_{t+1|t}^{(i)}, \tag{5a}$$

$$\boldsymbol{\Sigma}^{\boldsymbol{\eta}\mathbf{y}} = \mathbb{E}_i \left[(\mathbf{H}_{t+1|t} - \mathbb{E}_i[\mathbf{H}_{t+1|t}])^\top (\widehat{\mathbf{Y}} - \mathbb{E}_i[\widehat{\mathbf{Y}}]) \right]$$
$$= \frac{1}{N-1} \sum_{i=1}^N (\boldsymbol{\eta}_{t+1|t}^{(i)} - \overline{\boldsymbol{\eta}}_{t+1|t})^\top (\widehat{\mathbf{y}}^{(i)} - \mathbf{y}) \tag{5b}$$

After that, the forecast error covariance matrix of the predictions is computed

$$\boldsymbol{\Sigma}^{\mathbf{y}\mathbf{y}} = \mathbb{E}_i \left[(\widehat{\mathbf{Y}} - \mathbb{E}_i[\widehat{\mathbf{Y}}])^\top (\widehat{\mathbf{Y}} - \mathbb{E}_i[\widehat{\mathbf{Y}}]) \right]$$
$$= \frac{1}{N-1} \sum_{i=1}^N (\widehat{\mathbf{y}}^{(i)} - \mathbf{y})^\top (\widehat{\mathbf{y}}^{(i)} - \mathbf{y}) \tag{6}$$

Then the Kalman gain for correcting parameters can be computed as

$$\mathbf{K}^{\boldsymbol{\eta}} = \boldsymbol{\Sigma}^{\boldsymbol{\eta}\mathbf{y}} (\boldsymbol{\Sigma}^{\mathbf{y}\mathbf{y}} + \sigma_{\text{obs}}^2 \mathbf{I})^{-1} \tag{7}$$

The parameters are updated as

$$\boldsymbol{\eta}_{t+1|t+1}^{(i)} = \boldsymbol{\eta}_{t+1|t}^{(i)} + \mathbf{K}^{\boldsymbol{\eta}} (\mathbf{y}^{(i)} - \widehat{\mathbf{y}}^{(i)}) \tag{8}$$

4) Update state: Updates for state are similar to the updates for parameters, but with the updated values of parameters.

First, predictions of observations are corrected with updated parameters using

$$\widehat{\mathbf{y}}^{(i)} = K(\mathbf{X}_{\text{new}}, \mathbf{X}_g | \boldsymbol{\theta}_{t+1|t+1}^{(i)}) \times [K(\mathbf{X}_g, \mathbf{X}_g | \boldsymbol{\theta}_{t+1|t+1}^{(i)}) + \sigma_{y\,t+1|t+1}^{2(i)} \mathbf{I}]^{-1} \mathbf{g}_{t+1|t}^{(i)} \tag{9}$$

After that, the cross covariance of state ensemble and prediction ensemble is updated

$$\overline{\mathbf{g}}_{t+1|t} = \frac{1}{N} \sum_{i=1}^N \mathbf{g}_{t+1|t}^{(i)}, \tag{10a}$$

$$\boldsymbol{\Sigma}^{\mathbf{g}\mathbf{y}} = \mathbb{E}_i \left[(\mathbf{G}_{t+1|t} - \mathbb{E}[\mathbf{G}_{t+1|t}])^\top (\widehat{\mathbf{Y}} - \mathbb{E}[\widehat{\mathbf{Y}}]) \right]$$
$$= \frac{1}{N-1} \sum_{i=1}^N (\mathbf{g}_{t+1|t}^{(i)} - \overline{\mathbf{g}}_{t+1|t})^\top (\widehat{\mathbf{y}}^{(i)} - \mathbf{y}) \tag{10b}$$

After that, the forecast error covariance matrix of the predictions is computed according to (6) and then Kalman gain for correcting state

$$\mathbf{K}^{\mathbf{g}} = \boldsymbol{\Sigma}^{\mathbf{g}\mathbf{y}} (\boldsymbol{\Sigma}^{\mathbf{y}\mathbf{y}} + \sigma_{\text{obs}}^2 \mathbf{I})^{-1} \tag{11}$$

Then state is updated as

$$\mathbf{g}_{t+1|t+1}^{(i)} = \mathbf{g}_{t+1|t}^{(i)} + \mathbf{K}^{\mathbf{g}} (\mathbf{y}^{(i)} - \widehat{\mathbf{y}}^{(i)}) \tag{12}$$

The resulting algorithm is given in Fig.1.

B. Liu-West filter

The evolution of parameters distribution in (2) leads to over-diffuse of parameters distribution. Liu-West filter [25] uses kernel density estimation, it can be used to estimate the predicted distribution of parameters so that the resulting distribution converges to the true distribution. It is parametrised with discount factor $\delta_{lw} \in (0, 1]$, that is usually taken from the interval $[0.95, 0.99]$. With the introduction of additional parameters

$$a_{lw} = \frac{3\delta_{lw} - 1}{2\delta_{lw}}, \tag{13a}$$

$$h_{lw}^2 = 1 - a_{lw}^2, \tag{13b}$$

the evolution of parameters density is

$$\boldsymbol{\eta}_{t+1|t}^{(i)} = a_{lw} \boldsymbol{\eta}_{t|t}^{(i)} + (1 - a_{lw}) \overline{\boldsymbol{\eta}}_{t|t} + \boldsymbol{\varepsilon}_{lw}, \tag{14}$$

where $\boldsymbol{\varepsilon}_{lw} \sim \mathcal{N}(\mathbf{0}, \sqrt{h_{lw}^2 \operatorname{Var} \boldsymbol{\eta}_{t|t}})$ The algorithm is further denoted as Liu-West Dual GP-EnKF.

C. Joint Ensemble Kalman Filter for Gaussian Processes

It is also possible to estimate parameters of the model by augmenting the state vector $\mathbf{g}$ by parameter vector $\boldsymbol{\eta}$: augmented state is $\boldsymbol{s} = [\mathbf{g}; \boldsymbol{\eta}]$. The algorithm is further denoted as Joint GP-EnKF. The details are presented in Fig.2.

1) Initialise (1)
2) Iterate
  a) Predict, for each $1 \leq i \leq N$:
  P1 Predict parameters and state (2)
  P2 Predict observations (3)
  P3 Compute noisy trajectories (4)
  b) Update parameters
  UP1 Compute cross covariance of parameter ensemble and prediction ensemble (5)
  UP2 Compute forecast error covariance matrix of the predictions (6)
  UP3 Compute Kalman gain for correcting parameters (7)
  UP4 Update parameters (8)
  c) Update model state
  US1 Predict observations with updated parameters (9)
  US2 Compute cross covariance of state ensemble and prediction ensemble (10)
  US3 Compute forecast error covariance matrix of the predictions (6)
  US4 Compute Kalman gain for correcting state (11)
  US5 Update state (12)

Fig. 1: Dual GP-EnKF algorithm

1) Initialise (15)
2) Iterate
  a) Predict, for each $1 \leq i \leq N$:
  P1 Predict augmented state (16)
  P2 Predict observations (3)
  P3 Compute noisy trajectories (4)
  b) Update augmented state
  US1 Compute cross covariance of augmented state ensemble and prediction ensemble (17)
  US2 Compute forecast error covariance matrix of the predictions (6)
  US3 Compute Kalman gain for correcting augmented state (18)
  US4 Update augmented state (19)

Fig. 2: Joint GP-EnKF algorithm

After that, forecast error covariance matrix of the predictions is computed as (6) and then Kalman gain for correcting augmented state

$$\mathbf{K^s} = \mathbf{\Sigma^{sy}}(\mathbf{\Sigma^{yy}} + \sigma_y \mathbf{I})^{-1} \qquad (18)$$

Then augmented state is updated as

$$\mathbf{s}_{t+1|t+1}^{(i)} = \mathbf{s}_{t+1|t}^{(i)} + \mathbf{K^s}(\mathbf{y}^{(i)} - \widehat{\mathbf{y}}^{(i)}) \qquad (19)$$

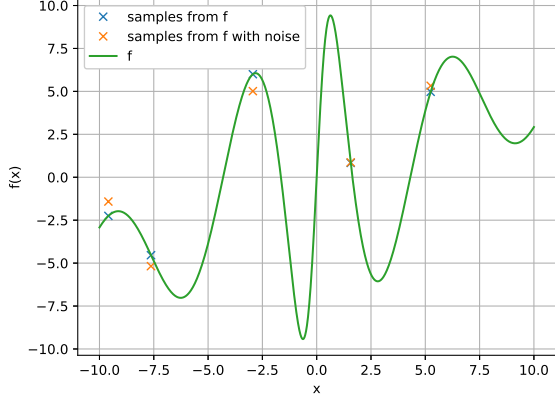*D. Computational complexity of Kalman Filter approaches for Gaussian Processes*

a) Dual Ensemble Kalman Filter: At the prediction step the most demanding operation is prediction of observations, that requires inversion of covariance matrix for each ensemble member, that is $\mathcal{O}(NK^3)$. At the update steps it is computation of Kalman gains, that is $\mathcal{O}(S^3) + \mathcal{O}(LS^2)$ for parameters and $\mathcal{O}(S^3) + \mathcal{O}(KS^2)$ for state. The resulting computational time complexity for the Dual GP-EnKF is $\mathcal{O}(T(NK^3 + S^3 + (L + K)S^2))$

b) Joint Ensemble Kalman Filter: Joint EnKF has the same asymptotic complexity as Dual EnKF.

c) Classic GP: The classic GP without inducing points that stores all previous observations and recomputes predictions at every time step t has $\mathcal{O}(S^3 t^3)$ computational complexity due to the covariance matrix growing in size as St at every dimension. The resulting computational complexity for the classic GP is then $\mathcal{O}(S^3 T^3)$. Note that the computational complexity of the dual ensemble Kalman Filter is linear with respect to the number T of time steps.

IV. EXPERIMENTS

In this section the performance of the proposed algorithms is evaluated on both synthetic and real data. Three versions of the EnKF for online GP parameters estimation are assessed: Dual GP-EnKF (Section III-A), Liu-West Dual GP-EnKF (Section III-B), and Joint EnKF (Section III-C). The developed algorithms are compared with the classic GP regression in

1) Initialisation: Initially, an ensemble for augmented state $\mathbf{S} \in \mathbb{R}^{N \times (L+K)} = [\mathbf{s}^{(i)}]_{1 \leq i \leq N}$ is generated. For each $1 \leq i \leq N$

$$\boldsymbol{\eta}_{0|0}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_S) \qquad (15a)$$

where $\mathbf{\Sigma}_S$ is the initial covariance matrices for the ensembles. After the initialisation the algorithm iterates prediction and update steps.

2) Prediction: Similar to the dual EnKF, the random walk assumption for motion model of the augmented state is assumed. Each ensemble member is updated as

$$\mathbf{s}_{t+1|t}^{(i)} = \mathbf{s}_{t|t}^{(i)} + \boldsymbol{\varepsilon}_s, \qquad (16)$$

where $\boldsymbol{\varepsilon}_s \sim \mathcal{N}(\mathbf{0}, \sigma_s \mathbf{I})$ is the noise variable with corresponding variance.

The predictions are made in a same way as in (3) and observations are noised as in (4).

3) Update: Updates for the augmented state are similar to the updates for state in dual EnKF. The cross covariance of augmented state ensemble and prediction ensemble is estimated as

$$\bar{\mathbf{s}}_{t+1|t} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{s}_{t+1|t}^{(i)}, \qquad (17a)$$

$$\mathbf{\Sigma^{sy}} = \mathbb{E}_i \left[(\mathbf{S}_{t+1|t} - \mathbb{E}[\mathbf{S}]_{t+1|t})^{\top} (\widehat{\mathbf{Y}} - \mathbb{E}[\widehat{\mathbf{Y}}]) \right]$$
$$= \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{s}_{t+1|t}^{(i)} - \bar{\mathbf{s}}_{t+1|t})^{\top} (\widehat{\mathbf{y}}^{(i)} - \mathbf{y}) \qquad (17b)$$
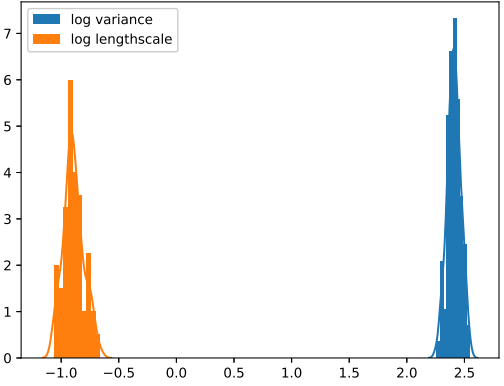
(a) The target function used in the synthetic data experiment. As an example, the sample from function used as the input for the algorithm at one iteration is displayed.
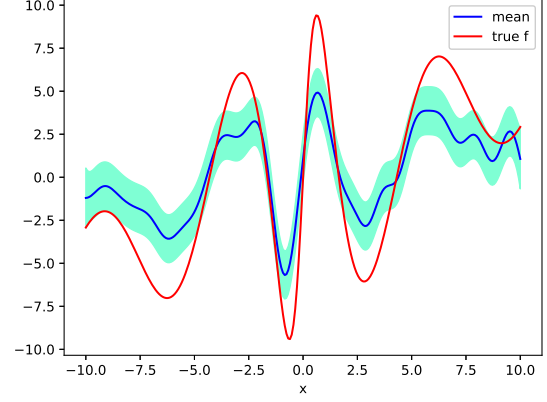
(a) Estimated mean, two standard deviation interval of the GP and target function
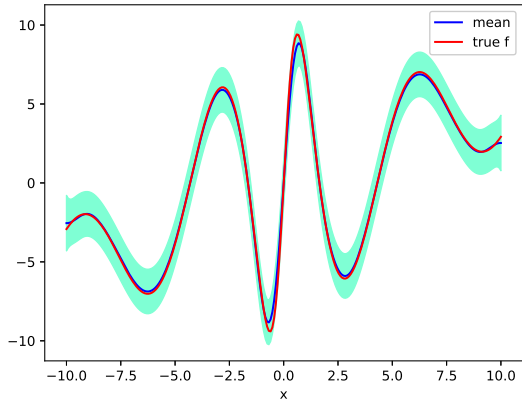
(b) The classic GP mean with two standard deviations. Parameters are optimised on the full history of observations.

Fig. 3: Target function and classic GP approximation

(b) Final distributions for the estimated GP parameters

Fig. 4: Performance of Joint GP-EnKF on the synthetic data

terms of both computational time and predictive accuracy. At every iteration t the classic GP regression is applied on all historical data.

For quantitative evaluation of the predictive accuracy, the normalised mean squared error of predictions (NMSE) is used on held-out test data $[\mathbf{x}^{\text{test}}, \mathbf{y}^{\text{test}}]$:

$$\text{NMSE} = \frac{1}{M} \sum_{m=1}^{M} \frac{\sqrt{(y_m^{\text{test}} - f^*(\mathbf{x}_m^{\text{test}}))^2}}{|y_m^{\text{test}}|}, \qquad (20)$$

where y_m^{test} is the observed value of the function at the test data point $\mathbf{x}_m^{\text{test}}$, $f^*(\mathbf{x}_m^{\text{test}})$ is the predicted function value at the test data point.

A. Synthetic data

The algorithms are firstly evaluated on the synthetic one-dimensional data. The target function for the synthetic data is

$$f(x) = \frac{x}{2} + \frac{25x}{1 + x^2} \cos(x) \qquad (21)$$

The observations are generated on the domain $[-10, 10]$ and are corrupted by the Gaussian noise with variance $\sigma_y^2 = 0.01$. The example of generated observations is presented in Fig. 3a.

All three versions of GP-EnKF use the size of the grid $K = 51$, and ensembles contain $N = 100$ members. The covariance function is squared-exponential [1], it has two hyperparameters: $\boldsymbol{\theta} = [\theta_{\text{variance}}, \theta_{\text{lengthscale}}]$ and estimates covariance between two points x_1 and x_2 as

$$K(x_1, x_2) = \theta_{\text{variance}} \exp \left\{ -\frac{||x_1 - x_2||_2^2}{\theta_{\text{lengthscale}}^2} \right\} \qquad (22)$$

At every iteration, $S = 5$ samples are fed into the algorithms, the total number of iterations is $T = 200$.

Fig. 3b shows the function estimate given by the classic GP regression. Since the total number of observations is sufficiently large, the classic GP is enable to reconstruct ideal predictions of the function.
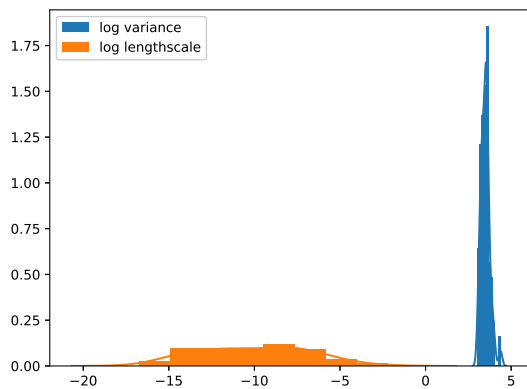
The performance of the proposed approaches is given in Fig. 4–6. Joint GP-EnKF (Fig. 4) correctly estimates peaks
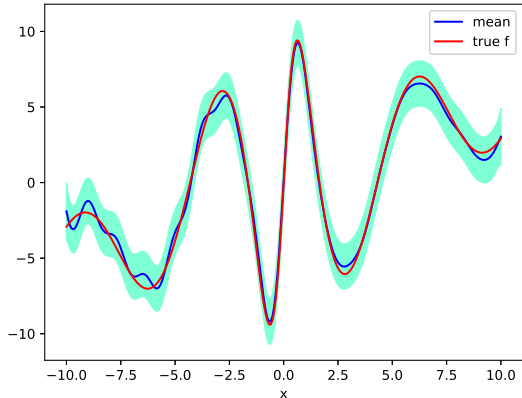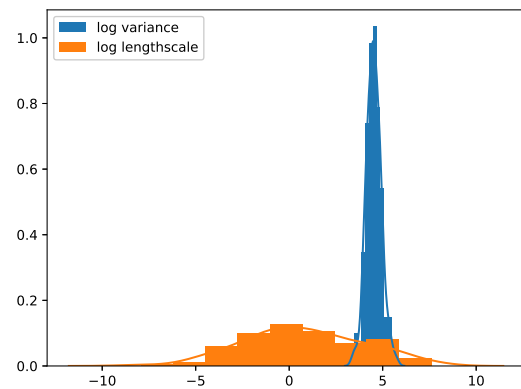
(a) Estimated mean, two standard deviation interval of the GP prediction and target function
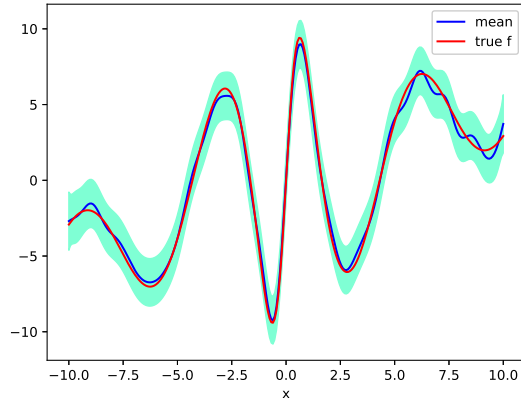
(b) Final distributions for the estimated GP parameters

Fig. 5: Performance of Dual GP-EnKF on the synthetic data

(a) Estimated mean, two standard deviation interval of the GP and target function

(b) Final distributions for the estimated GP parameters

Fig. 6: Performance of the Liu-West Dual GP-EnKF on the synthetic data

of the target function, but it has large predictive errors for most of the observations. Joint GP-EnKF learns the consistent ensemble estimates of the hyperparameters, i.e. their variance is not large.

Dual GP-EnKF (Fig. 5) provides predictions that are more accurate that predictions made by Joint GP-EnKF, but still there are several locations where the shape of the target function differs from the predicted mean. The ensemble of Dual GP-EnKF has low variance for the logarithm of the lengthscale hyperparameter of the covariance function and high variance for the estimates of the variance hyperparameter.

Liu-West Dual GP-EnKF (Fig. 6) is applied with discount factor $\delta_{lw} = 0.95$. The algorithm makes predictions that are closer to the true values of the target function than other algorithms. The ensemble of Liu-West Dual GP-EnKF gives better estimations of hyperparameters than both Dual and Joint GP-EnKFs.

The procedure is repeated for 10 Monte Carlo runs with different random seeds. The results are presented as average among these 10 Monte Carlo runs. In the Fig.7 the history

of NMSE is given over time. It final values together with the computational time are presented in Table I. While Joint GP-EnKF is the fastest method, NMSE of both Dual GP-EnKF methods is lower, and Liu-West Dual GP-EnKF provides the best results. The classic GP approach provides the lowest NMSE, however, it has the computational time more than 10 times higher than of the slowest of the proposed approaches. In terms of the likelihood all methods show similar results with Joint GP-EnKF slightly outperforming the other two algorithms.

TABLE I: Performance on the synthetic data at $T = 200$

Method	NMSE	Time (s)
Joint GP-EnKF	0.64	**7.23**
Dual GP-EnKF	0.48	13.68
Liu-West Dual GP-EnKF	**0.19**	15.60
classic GP	0.02	186.20

Fig. 7: History of NMSE

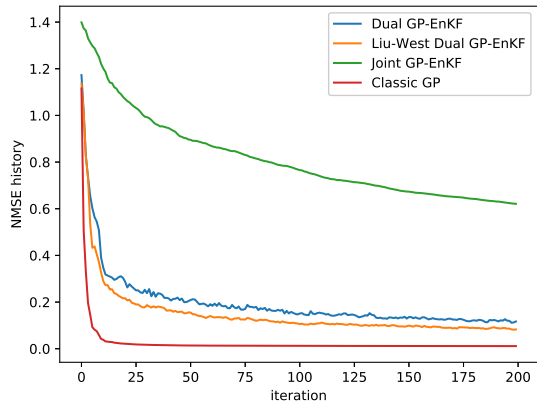

(a) Mean prices after the first batch of data is presented to the algorithm

B. House Prices

The Dual GP-EnKF approach is further evaluated on the real data. In this example the HM Land Registry Price Paid Data[1] is considered. The subset of all flats and maisonettes sold in 2017 is selected and the parameters estimation is performed to predict mean prices corresponding to the locations of properties. Longitude and latitude values for every location have been calculated based on the postcode. Therefore, in this experiment, every single input $\mathbf{x}$ is two-dimensional.

A total of $T = 20$ iterations have been performed with two-dimensional grid of size $K = 25 \times 25 = 625$. At every iteration, $S = 100$ samples of the logarithms of standardised prices are used to update parameters and mean in the grid points. The ensemble consists of $N = 200$ members. The covariance function is stationary squared-exponential.

Fig.8 demonstrates the results after the first and final iterations. It is clear that the prices have converged close to real values, identifying such areas as London and Oxford as places with higher prices. Though there are spikes of the mean in the sea, the corresponding covariance values that describe uncertainty of predictions in these points are high. Note that the used squared-exponential covariance function is one of the simplest covariance functions in terms of complexity of modelling dependencies of function values at different data points. The stationary squared-exponential covariance function does not depend on locations. Therefore, the results can potentially be further improved if the squared-exponential covariance function is considered together with non-stationary covariance functions to obtain more precise estimates for covariance difference between sea and land locations.
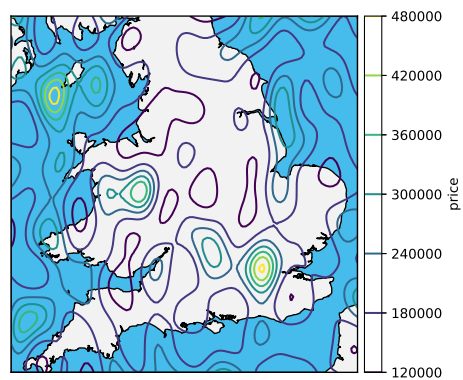
V. CONCLUSIONS

The paper proposes two ensemble Kalman filters for online Gaussian process regression and learning. The mean and hyperparameters of the GP are interpreted as the state and parameters of the ensemble Kalman filter, respectively. The
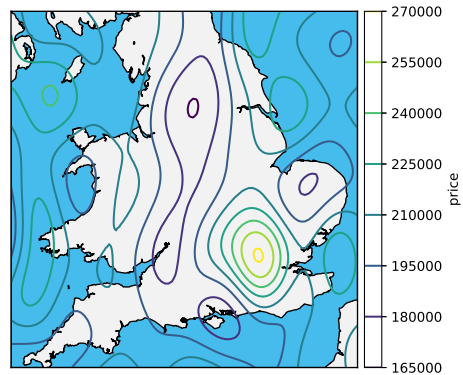
(b) Mean prices after the 20th batch of data is presented to the algorithm

Fig. 8: Mean estimates of the prices with Dual GP-EnKF

ensemble Kalman filter updates are utilised to recursively improve estimates of both state and parameters. Two versions of the ensemble updates are proposed: Joint GP-EnKF where the update step of the EnKF is applied for the augmented vector-parameter state and Dual GP-EnKF where the update step is split to first update the parameters and then based on new estimates of the parameters the state is updated. For the Dual EnKF the Liu-West filter ([25]) updates are additionally developed for further improvement of the estimates.

The proposed ensemble Kalman filter approach for the GP has a linear computational complexity with respect to the number of sequential observations, it depends mainly on the dimensionality of the observations at each timestamp and internal parameters of the filter. For the large volume of data acquired sequentially, it can significantly reduce the computational time in comparison to the usual GP regression that scales cubically with respect to the number of observations. Starting from sufficient number of observations, cubic complexity makes the usual GP not applicable for this large-scale data. The proposed ensemble Kalman filter can be used

[1] https://data.gov.uk/dataset/land-registry-monthly-price-paid-data/

with any number of sequential observations given that at each timestamp the dimensionality of observations is feasible.

The experiments both on synthetic and real data show that the proposed ensemble Kalman filter approaches for the Gaussian process estimation provide satisfactory predictive accuracy using significantly less computational time in comparison to the GP regression without online updates. Among the proposed approaches the Liu-West Dual GP-EnKF filter demonstrates the best results in terms of the predictive accuracy slightly underperforming the Joint EnKF in terms of the computational time.

REFERENCES

[1] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[2] R. E. Turner and M. Sahani, "Demodulation as probabilistic inference," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2398–2411, 2011.

[3] F. Pérez-Cruz, S. Van Vaerenbergh, J. J. Murillo-Fuentes, M. Lázaro-Gredilla, and I. Santamaria, "Gaussian processes for nonlinear signal processing: An overview of recent advances," *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 40–50, 2013.

[4] A. Svensson, J. Dahlin, and T. B. Schön, "Marginalizing Gaussian process hyperparameters using sequential Monte Carlo," in *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 2015, pp. 477–480.

[5] J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *Journal of Machine Learning Research*, vol. 6, no. Dec, pp. 1939–1959, 2005.

[6] M. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *Artificial Intelligence and Statistics*, 2009, pp. 567–574.

[7] T. D. Bui, J. Yan, and R. E. Turner, "A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation," *Journal of Machine Learning Research*, vol. 18, no. 104, pp. 1–72, 2017.

[8] Y. Shen, M. Seeger, and A. Y. Ng, "Fast Gaussian process regression using KD-trees," in *Advances in neural information processing systems*, 2006, pp. 1225–1232.

[9] Y. Gal, M. Van Der Wilk, and C. E. Rasmussen, "Distributed variational inference in sparse Gaussian process regression and latent variable models," in *Advances in Neural Information Processing Systems*, 2014, pp. 3257–3265.

[10] M. F. Huber, "Recursive Gaussian process: On-line regression and learning," *Pattern Recognition Letters*, vol. 45, pp. 85–91, 2014.

[11] F. Yin, Y. Zhao, F. Gunnarsson, and F. Gustafsson, "Received-signal-strength threshold optimization using Gaussian processes," *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 2164–2177, 2017.

[12] F. Yin and F. Gunnarsson, "Distributed recursive Gaussian processes for RSS map applied to target tracking," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 3, pp. 492–503, 2017.

[13] Y. Zhao, F. Yin, F. Gunnarsson, F. Hultkratz, and J. Fagerlind, "Gaussian processes for flow modeling and prediction of positioned trajectories evaluated with sports data," in *19th International Conference on Information Fusion (FUSION)*. IEEE, 2016, pp. 1461–1468.

[14] I. Murray and R. P. Adams, "Slice sampling covariance hyperparameters of latent Gaussian models," in *Advances in Neural Information Processing Systems*, 2010, pp. 1732–1740.

[15] M. A. Osborne, S. J. Roberts, A. Rogers, and N. R. Jennings, "Real-time information processing of environmental sensor network data using bayesian Gaussian processes," *ACM Transactions on Sensor Networks (TOSN)*, vol. 9, no. 1, p. 1, 2012.

[16] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics," *Journal of Geophysical Research: Oceans*, vol. 99, no. C5, pp. 10 143–10 162, 1994.

[17] M. Roth, G. Hendeby, C. Fritsche, and F. Gustafsson, "The ensemble Kalman filter: a signal processing perspective," *EURASIP Journal on Advances in Signal Processing*, vol. 2017, no. 1, p. 56, 2017.

[18] J. L. Anderson, "An ensemble adjustment Kalman filter for data assimilation," *Monthly weather review*, vol. 129, no. 12, pp. 2884–2903, 2001.

[19] G. Evensen, "The ensemble Kalman filter for combined state and parameter estimation," *IEEE Control Systems*, vol. 29, no. 3, 2009.

[20] E. A. Wan and A. T. Nelson, "Dual Kalman filtering methods for nonlinear prediction, smoothing and estimation," in *Advances in neural information processing systems*, 1997, pp. 793–799.

[21] H. Moradkhani, S. Sorooshian, H. V. Gupta, and P. R. Houser, "Dual state–parameter estimation of hydrological models using ensemble Kalman filter," *Advances in water resources*, vol. 28, no. 2, pp. 135–147, 2005.

[22] H. L. Mitchell and P. Houtekamer, "An adaptive ensemble Kalman filter," *Monthly Weather Review*, vol. 128, no. 2, pp. 416–433, 2000.

[23] T. DelSole and X. Yang, "State and parameter estimation in stochastic dynamical models," *Physica D: Nonlinear Phenomena*, vol. 239, no. 18, pp. 1781–1788, 2010.

[24] J. R. Stroud and T. Bengtsson, "Sequential state and variance estimation within the ensemble Kalman filter," *Monthly Weather Review*, vol. 135, no. 9, pp. 3194–3208, 2007.

[25] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 197–223.