Real-time Gaussian Process Regression

Bill Menke, June 11, 2022

Statement of the problem: Gaussian Process Regression (GPR) is used to estimate a set of model parameters that represent a discretized field using data and prior information about the autocovariance of the field. At every time increment, new data become available. We seek an efficient method for continually updating the estimate to include the new data. This process can be considered real-time GPR.

We use the formulation of Gaussian Process Regression (GPR) in which the model parameters $\mathbf{m} = \left[\mathbf{m}^{(t)};\ \mathbf{m}^{(c)}\right]$ are divided into target, t, and control, c, types, the latter of which are observed by data, $\mathbf{d}$. The target points might represent a gridded version of the field, and the control points observations of the field at irregularly-spaced locations. The prior covariance of the model parameters is $\mathbf{C}_m$ and the variance of the data is $\sigma_d^2\mathbf{I}$. The GPR estimate of the model parameters is then:

$$\Delta\mathbf{m} \equiv \begin{bmatrix}\Delta\mathbf{m}^{(t)}\\ \Delta\mathbf{m}^{(c)}\end{bmatrix} = \begin{bmatrix}\mathbf{C}_m^{(tc)}\\ \mathbf{C}_m^{(cc)}\end{bmatrix}\mathbf{u}$$

$$\text{with}\quad \mathbf{Mu} = \Delta\mathbf{d} \quad\text{and}\quad \mathbf{M} \equiv \mathbf{C}_m^{(cc)} + \sigma_d^2\mathbf{I}$$

$$\text{and}\quad \Delta\mathbf{m} \equiv \mathbf{m}^{est} - \mathbf{m}^{pri} \quad\text{and}\quad \Delta\mathbf{d} \equiv \mathbf{d}^{obs} - \mathbf{m}^{(c,pri)}$$

(1)

We now suppose that we know the vector, $\hat{\mathbf{u}}_\mathbf{1}$, when N_1 data, $\Delta\mathbf{d}_\mathbf{1}$, are available, and seek to understand how to calculate the vector, $\mathbf{u} \equiv [\mathbf{u}_\mathbf{1};\ \mathbf{u}_\mathbf{2}]$, when an additional N_2 data are provided:

$$[\mathbf{A}][\hat{\mathbf{u}}_\mathbf{1}] = [\Delta\mathbf{d}_\mathbf{1}] \quad\text{and}\quad \begin{bmatrix}\mathbf{A} & \mathbf{C}^T\\ \mathbf{C} & \mathbf{B}\end{bmatrix}\begin{bmatrix}\mathbf{u}_\mathbf{1}\\ \mathbf{u}_\mathbf{2}\end{bmatrix} = \begin{bmatrix}\Delta\mathbf{d}_\mathbf{1}\\ \Delta\mathbf{d}_\mathbf{2}\end{bmatrix}$$

$$\Delta\hat{\mathbf{m}}^{(t)} = \mathbf{C}_m^{(tc1)}\hat{\mathbf{u}}_\mathbf{1} \quad\text{and}\quad \Delta\mathbf{m}^{(t)} = \begin{bmatrix}\mathbf{C}_m^{(tc1)} & \mathbf{C}_m^{(tc2)}\end{bmatrix}\begin{bmatrix}\mathbf{u}_\mathbf{1}\\ \mathbf{u}_\mathbf{2}\end{bmatrix}$$

$$\Delta\hat{\mathbf{m}}^{(c)} = \left\{\mathbf{A} - \sigma_d^2\mathbf{I}\right\}\hat{\mathbf{u}}_\mathbf{1} \quad\text{and}\quad \Delta\mathbf{m}^{(c)} = \begin{bmatrix}\Delta\mathbf{m}^{(c1)}\\ \Delta\mathbf{m}^{(c2)}\end{bmatrix} = \begin{bmatrix}\mathbf{A} & \mathbf{C}^T\\ \mathbf{C} & \mathbf{B}\end{bmatrix}\begin{bmatrix}\mathbf{u}_\mathbf{1}\\ \mathbf{u}_\mathbf{2}\end{bmatrix} - \sigma_d^2\begin{bmatrix}\mathbf{u}_\mathbf{1}\\ \mathbf{u}_\mathbf{2}\end{bmatrix}$$

(2)

Here, the variables with hats refer to the solution when N_1 are available, and those without hats refer to when $N_1 + N_2$ are available. The sub-matrices are defined as $\mathbf{A} \equiv \mathbf{C}_m^{(c1c1)} + \sigma_d^2\mathbf{I}$, $\mathbf{B} \equiv \mathbf{C}_m^{(c2c2)} + \sigma_d^2\mathbf{I}$ and $\mathbf{C} \equiv \mathbf{C}_m^{(c1c2)}$, where the superscript $c1$ denotes the N_1 control points and $c2$ denotes the N_2 control points. The bordering method can be used to show that

$$\begin{bmatrix}\mathbf{A} & \mathbf{C}^T\\ \mathbf{C} & \mathbf{B}\end{bmatrix}^{-1} = \begin{bmatrix}\mathbf{D} & \mathbf{F}^T\\ \mathbf{F} & \mathbf{E}\end{bmatrix}$$

(3)

with

$$\mathbf{D} = \mathbf{Z}\mathbf{A}^{-1} \quad \text{with} \quad \mathbf{Z} \equiv [\mathbf{I} + \mathbf{A}^{-1}\mathbf{C}^T[\mathbf{B} - \mathbf{C}\mathbf{A}^{-1}\mathbf{C}^T]^{-1}\mathbf{C}]$$
$$\mathbf{E} = [\mathbf{B} - \mathbf{C}\mathbf{A}^{-1}\mathbf{C}^T]^{-1}$$
$$\mathbf{F} = -[\mathbf{B} - \mathbf{C}\mathbf{A}^{-1}\mathbf{C}^T]^{-1}\mathbf{C}\mathbf{A}^{-1} \tag{4}$$

We now introduce the quantities

$$\mathbf{U} \quad \text{such that} \quad \mathbf{U} = \mathbf{A}^{-1}\mathbf{C}^T$$
$$\mathbf{Y} \quad \text{such that} \quad [\mathbf{B} - \mathbf{C}\mathbf{U}]\mathbf{Y} = \mathbf{C}$$
$$\hat{\mathbf{u}}_2 \quad \text{such that} \quad \mathbf{B}\hat{\mathbf{u}}_2 = \Delta\mathbf{d}_2 \tag{5}$$

These definitions imply:

$$\mathbf{D} = [\mathbf{I} + \mathbf{U}Y]\mathbf{A}^{-1}$$
$$\mathbf{E} = [\mathbf{B} - \mathbf{C}\mathbf{U}]^{-1}$$
$$\mathbf{F} = -\mathbf{E}\mathbf{U}^T \tag{6}$$

Then the solution for $\mathbf{u} \equiv [\mathbf{u}_1;\ \mathbf{u}_2]$ is:

$$\mathbf{u}_1 = [\mathbf{I} + \mathbf{U}Y][\hat{\mathbf{u}}_1 - \mathbf{U}\hat{\mathbf{u}}_2]$$
$$[\mathbf{B} - \mathbf{C}\mathbf{U}]\mathbf{u}_2 = \Delta\mathbf{d}_2 - \mathbf{C}\hat{\mathbf{u}}_1 \tag{7}$$

These formulas have been tested numerically. The recursion for the solution is then:

Step 1:

Set $K = 1$

Set N_1 to the number of data available for the Set $K = 1$ time step

Calculate the matrices, $\mathbf{A} = \mathbf{C}_m^{(c1c1)} + \sigma_d^2\mathbf{I}$ and $\mathbf{C}_m^{(tc1)}$

Compute $\mathbf{A}^{-1}$ and $\hat{\mathbf{u}}_1 = \mathbf{A}^{-1}\Delta\mathbf{d}_1$

Solve for $\Delta\hat{\mathbf{m}}^{(t)}$ and $\Delta\hat{\mathbf{m}}^{(c)}$ as in Eqn (2)

Step 2:

Increment K

Set N_2 to the number of new data

Calculate the submatrices $\mathbf{B} = \mathbf{C}_m^{(c2c2)} + \sigma_d^2 \mathbf{I}$, $\mathbf{C} = \mathbf{C}_m^{(c1c2)}$ and $\mathbf{C}_m^{(tc2)}$

Solve for $\mathbf{U}$, $\mathbf{Y}$, and $\hat{\mathbf{u}}_2$ as in Eqn (5)

Solve for $\mathbf{u}_1$ and $\mathbf{u}_2$ as in Eqn (7)

Perform multiplications to get $\Delta\mathbf{m}^{(t)}$ and $\Delta\mathbf{m}^{(c)}$ as in Eqn (2)

Step 3:

$\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}$ becomes $\hat{\mathbf{u}}_1$

$\begin{bmatrix} \mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{B} \end{bmatrix}$ becomes $\mathbf{A}$

$\begin{bmatrix} \mathbf{D} & \mathbf{F}^T \\ \mathbf{F} & \mathbf{E} \end{bmatrix}$ becomes $\mathbf{A}^{-1}$

Refine $\mathbf{A}^{-1}$ as described in the note, below

$\begin{bmatrix} \mathbf{C}_m^{(tc1)} & \mathbf{C}_m^{(tc2)} \end{bmatrix}$ becomes $\mathbf{C}_m^{(tc1)}$

$N_1 + N_2$ becomes N_1

Go to Step 2

Note that the effort needed to update the solution grows with each iteration, because the size of **U** and **Y** continually increases. Furthermore, the rate of growth is faster than linear in time. This in in contrast to Kalman filtering, where the effort is constant. The difference arises from sparsity. The fundamental matrix of Kalman filtering is sparse; the matrix, **A**, is not.

Note: I have found necessary refining $\mathbf{A}^{-1}$ after each application of the bordering method, using an iterative first-order correction scheme. Assume that the $\mathbf{A}^{-1}$ resulting from bordering – say $\mathbf{A}'^{-1}$ - is inaccurate by a small amount $\Delta\mathbf{D}$, so that $\mathbf{A} = [\mathbf{A}'^{-1} + \Delta\mathbf{D}]^{-1}$. Then, we use the first order rule $[\mathbf{A}'^{-1} + \Delta\mathbf{D}]^{-1} = [\mathbf{A}'^{-1}]^{-1} - [\mathbf{A}'^{-1}]^{-1}\Delta\mathbf{D}[\mathbf{A}'^{-1}]^{-1}$ to write $\mathbf{A} = [\mathbf{A}'^{-1}]^{-1} - [\mathbf{A}'^{-1}]^{-1}\Delta\mathbf{D}[\mathbf{A}'^{-1}]^{-1}$. Solving for $\Delta\mathbf{D}$ yields $\Delta\mathbf{D} = \mathbf{A}'^{-1} - \mathbf{A}'^{-1}\mathbf{A}\mathbf{A}'^{-1}$ and an improved estimate of the inverse is $\mathbf{A}^{-1} = \mathbf{A}'^{-1} + \Delta\mathbf{D}$. A few (sat, 3) iterations of this procedure correct errors introduced by successive application of bordering.

Example: The goal of this numerical experiment is to reconstruct a two-dimensional field, $m(x, y)$, on the interval $0 \leq x \leq 1$, $0 \leq y \leq 1$, evaluated on a 30×30 grid of uniformly-spaced target points. At each time step, a total of 5 data are collected, drawn at randomly chosen points from the true function $m(x, y) = \sin(2\pi x)\sin(2\pi y)$ and with variance $\sigma_d^2 = 0.01$ The field is assumed to have the Gaussian autocovariace, $C(x, x', y, y') = \exp(-½ r^2/s^2)$, with $r^2 = (x - x')^2 + (y - y')^2$ and scale length $s = 0.22$. The reconstruction systematically improves with time, as new data are obtained (Fig. 1).

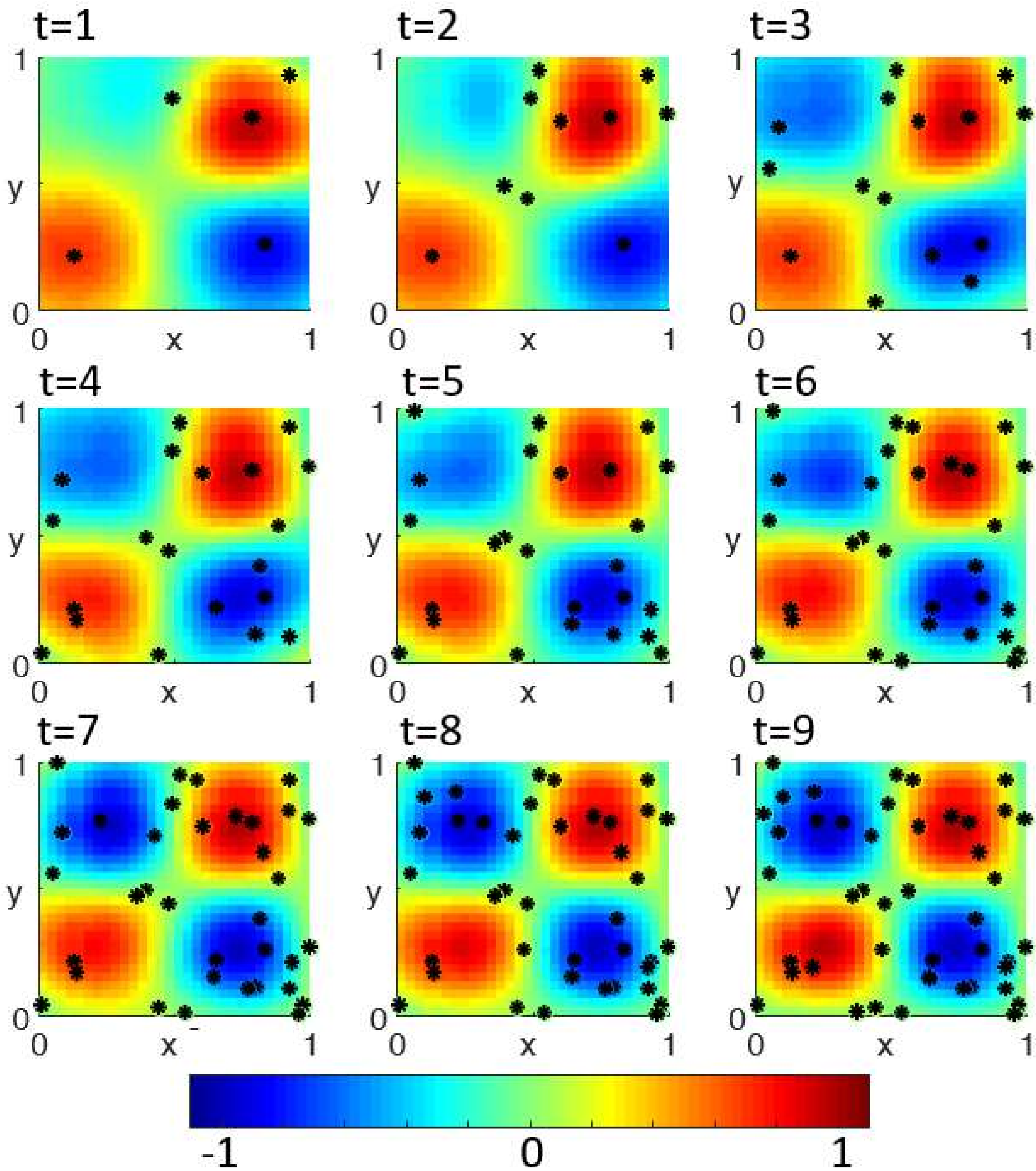

Figure 1. The sinusoidal field, $m(x, y)$, reconstructed at a sequence of 9 time steps. Five data are added in each time step.