Bayesian smoothing with extended second order random walk model: An detailed overview and comparison

Ziang Zhang

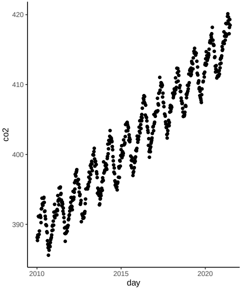
Supervisors: Patrick Brown
James Stafford

Department of Statistics, University of Toronto

November 2021

Outline

Consider the atmospheric Carbon Dioxide (CO2) concentrations data from an observatory in Hawaii. This dataset contains the observation of CO2 concentrations from 1960 to 2021, with unequally spaced observation times.



Model:

$$y_i = f_p(t_i) + f_{np}(t_i) + \epsilon_i$$

- $\epsilon_i \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$
- $f_p(t_i)$ is defined as $\beta_0 + \beta_1 \cos(2\pi t) + \beta_2 \sin(2\pi t) + \beta_3 \cos(4\pi t) + \beta_4 \sin(4\pi t)$.
- $f_{np}(t_i)$ is the non-parametric part.

Question of interest: A smooth function $f_{np}(t)$ as well as its derivative for $t \geq 2020$.

Model:

$$y_i = f_p(t_i) + f_{np}(t_i) + \epsilon_i$$

- $\epsilon_i \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$
- $f_p(t_i)$ is defined as $\beta_0 + \beta_1 \cos(2\pi t) + \beta_2 \sin(2\pi t) + \beta_3 \cos(4\pi t) + \beta_4 \sin(4\pi t)$.
- $f_{np}(t_i)$ is the non-parametric part.

Question of interest: A smooth function $f_{np}(t)$ as well as its derivative for $t \geq 2020$.

Model:

$$y_i = f_p(t_i) + f_{np}(t_i) + \epsilon_i$$

- $\epsilon_i \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$
- $f_p(t_i)$ is defined as $\beta_0 + \beta_1 \cos(2\pi t) + \beta_2 \sin(2\pi t) + \beta_3 \cos(4\pi t) + \beta_4 \sin(4\pi t)$.
- $f_{np}(t_i)$ is the non-parametric part.

Question of interest: A smooth function $f_{np}(t)$ as well as its derivative for $t \geq 2020$.

Model:

$$y_i = f_p(t_i) + f_{np}(t_i) + \epsilon_i$$

- $\epsilon_i \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$
- $f_p(t_i)$ is defined as $\beta_0 + \beta_1 \cos(2\pi t) + \beta_2 \sin(2\pi t) + \beta_3 \cos(4\pi t) + \beta_4 \sin(4\pi t)$.
- $f_{np}(t_i)$ is the non-parametric part.

Question of interest: A smooth function $f_{np}(t)$ as well as its derivative for $t \geq 2020$.

Model:

$$y_i = f_p(t_i) + f_{np}(t_i) + \epsilon_i$$

- $\epsilon_i \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$
- $f_p(t_i)$ is defined as $\beta_0 + \beta_1 \cos(2\pi t) + \beta_2 \sin(2\pi t) + \beta_3 \cos(4\pi t) + \beta_4 \sin(4\pi t)$.
- $f_{np}(t_i)$ is the non-parametric part.

Question of interest: A smooth function $f_{np}(t)$ as well as its derivative for $t \geq 2020$.

Model:

$$y_i = f_p(t_i) + f_{np}(t_i) + \epsilon_i$$

- $\epsilon_i \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$
- $f_p(t_i)$ is defined as $\beta_0 + \beta_1 \cos(2\pi t) + \beta_2 \sin(2\pi t) + \beta_3 \cos(4\pi t) + \beta_4 \sin(4\pi t)$.
- $f_{np}(t_i)$ is the non-parametric part.

Question of interest: A smooth function $f_{np}(t)$ as well as its derivative for $t \geq 2020$.

Fitting Smoothing Spline

Consider $y_i = g(x_i) + \epsilon_i$ where $\epsilon_i \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$ and $x_i \in [a, b]$, then the (traditional) smoothing spline aims to solve:

$$\arg\min_{g \in C^2} \left\{ \sum_i \left(\frac{y_i - g(x_i)}{\sigma_\epsilon} \right)^2 + \lambda \int_a^b g''(x)^2 dx \right\} \tag{1}$$

The sum of square term on the left can be replaced by negative log likelihood, which is also called *penalized likelihood* method.

Problem: The parameters σ_ξ and λ are unknown.
One Solution: Bayesian hierarchical model.

Fitting Smoothing Spline

Consider $y_i = g(x_i) + \epsilon_i$ where $\epsilon_i \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$ and $x_i \in [a, b]$, then the (traditional) smoothing spline aims to solve:

$$\underset{g \in C^2}{\arg\min} \left\{ \sum_i \left(\frac{y_i - g(x_i)}{\sigma_\epsilon} \right)^2 + \lambda \int_a^b g''(x)^2 dx \right\} \tag{1}$$

The sum of square term on the left can be replaced by negative log likelihood, which is also called *penalized likelihood* method.

Problem: The parameters σ_ξ and λ are unknown.
One Solution: Bayesian hierarchical model.

Fitting Smoothing Spline

Consider $y_i = g(x_i) + \epsilon_i$ where $\epsilon_i \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$ and $x_i \in [a, b]$, then the (traditional) smoothing spline aims to solve:

$$\arg\min_{g \in C^2} \left\{ \sum_i \left(\frac{y_i - g(x_i)}{\sigma_\epsilon} \right)^2 + \lambda \int_a^b g''(x)^2 dx \right\} \tag{1}$$

The sum of square term on the left can be replaced by negative log likelihood, which is also called *penalized likelihood* method.

Problem: The parameters σ_ξ and λ are unknown.
One Solution: Bayesian hierarchical model.

Fitting Smoothing Spline

Consider $y_i = g(x_i) + \epsilon_i$ where $\epsilon_i \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$ and $x_i \in [a, b]$, then the (traditional) smoothing spline aims to solve:

$$\arg\min_{g \in C^2} \left\{ \sum_i \left(\frac{y_i - g(x_i)}{\sigma_\epsilon} \right)^2 + \lambda \int_a^b g''(x)^2 dx \right\} \tag{1}$$

The sum of square term on the left can be replaced by negative log likelihood, which is also called *penalized likelihood* method.

Problem: The parameters σ_ξ and λ are unknown.

One Solution: Bayesian hierarchical model.

Fitting Smoothing Spline

Consider $y_i = g(x_i) + \epsilon_i$ where $\epsilon_i \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$ and $x_i \in [a, b]$, then the (traditional) smoothing spline aims to solve:

$$\underset{g \in C^2}{\arg\min} \left\{ \sum_i \left(\frac{y_i - g(x_i)}{\sigma_\epsilon} \right)^2 + \lambda \int_a^b g''(x)^2 dx \right\} \tag{1}$$

The sum of square term on the left can be replaced by negative log likelihood, which is also called *penalized likelihood* method.

Problem: The parameters σ_ξ and λ are unknown.
One Solution: Bayesian hierarchical model.

Vectorize the equation 1 into the following:

$$\frac{1}{\sigma_\epsilon^2}(\boldsymbol{y}-\boldsymbol{g})^T(\boldsymbol{y}-\boldsymbol{g})+\lambda\boldsymbol{g}^T K\boldsymbol{g}, \tag{2}$$

This can be interpreted as:

$$\text{log-likelihood}+\text{prior for g.} \tag{3}$$

The matrix K can be factorized as the following:

$$K = D^T R^{-1} D. \tag{4}$$

The $(n-2)\times n$ matrix D is a second-order differencing matrix. The $(n-2)\times(n-2)$ matrix R^{-1} corresponds to the precision matrix of a MA(1) process (Brown and De Jong, 2001).

Vectorize the equation 1 into the following:

$$\frac{1}{\sigma_\epsilon^2}(\boldsymbol{y}-\boldsymbol{g})^T(\boldsymbol{y}-\boldsymbol{g})+\lambda\boldsymbol{g}^T K\boldsymbol{g}, \tag{2}$$

This can be interpreted as:

$$\text{log-likelihood}+\text{prior for g.} \tag{3}$$

The matrix K can be factorized as the following:

$$K = D^T R^{-1} D. \tag{4}$$

The $(n-2)\times n$ matrix D is a second-order differencing matrix. The $(n-2)\times(n-2)$ matrix R^{-1} corresponds to the precision matrix of a MA(1) process (Brown and De Jong, 2001).

Vectorize the equation 1 into the following:

$$\frac{1}{\sigma_\epsilon^2}(\boldsymbol{y}-\boldsymbol{g})^T(\boldsymbol{y}-\boldsymbol{g})+\lambda\boldsymbol{g}^TK\boldsymbol{g}, \tag{2}$$

This can be interpreted as:

$$\text{log-likelihood}+\text{prior for g.} \tag{3}$$

The matrix K can be factorized as the following:

$$K = D^T R^{-1} D. \tag{4}$$

The $(n-2)\times n$ matrix D is a second-order differencing matrix. The $(n-2)\times(n-2)$ matrix R^{-1} corresponds to the precision matrix of a MA(1) process (Brown and De Jong, 2001).

Vectorize the equation 1 into the following:

$$\frac{1}{\sigma_\epsilon^2}(\boldsymbol{y}-\boldsymbol{g})^T(\boldsymbol{y}-\boldsymbol{g})+\lambda\boldsymbol{g}^T K\boldsymbol{g}, \tag{2}$$

This can be interpreted as:

$$\text{log-likelihood}+\text{prior for g}. \tag{3}$$

The matrix K can be factorized as the following:

$$K = D^T R^{-1} D. \tag{4}$$

The $(n-2)\times n$ matrix D is a second-order differencing matrix. The $(n-2)\times(n-2)$ matrix R^{-1} corresponds to the precision matrix of a MA(1) process (Brown and De Jong, 2001).

Vectorize the equation 1 into the following:

$$\frac{1}{\sigma_\epsilon^2}(\boldsymbol{y}-\boldsymbol{g})^T(\boldsymbol{y}-\boldsymbol{g})+\lambda\boldsymbol{g}^T K\boldsymbol{g}, \tag{2}$$

This can be interpreted as:

$$\text{log-likelihood}+\text{prior for g}. \tag{3}$$

The matrix K can be factorized as the following:

$$K = D^T R^{-1} D. \tag{4}$$

The $(n-2)\times n$ matrix D is a second-order differencing matrix. The $(n-2)\times(n-2)$ matrix R^{-1} corresponds to the precision matrix of a MA(1) process (Brown and De Jong, 2001).

Vectorize the equation 1 into the following:

$$\frac{1}{\sigma_\epsilon^2}(\boldsymbol{y}-\boldsymbol{g})^T(\boldsymbol{y}-\boldsymbol{g})+\lambda\boldsymbol{g}^T K\boldsymbol{g}, \tag{2}$$

This can be interpreted as:

$$\text{log-likelihood}+\text{prior for g}. \tag{3}$$

The matrix K can be factorized as the following:

$$K = D^T R^{-1} D. \tag{4}$$

The $(n-2)\times n$ matrix D is a second-order differencing matrix. The $(n-2)\times(n-2)$ matrix R^{-1} corresponds to the precision matrix of a MA(1) process (Brown and De Jong, 2001).

Vectorize the equation 1 into the following:

$$\frac{1}{\sigma_\epsilon^2}(\boldsymbol{y}-\boldsymbol{g})^T(\boldsymbol{y}-\boldsymbol{g})+\lambda\boldsymbol{g}^T K\boldsymbol{g}, \tag{2}$$

This can be interpreted as:

$$\text{log-likelihood}+\text{prior for g}. \tag{3}$$

The matrix K can be factorized as the following:

$$K = D^T R^{-1} D. \tag{4}$$

The $(n-2)\times n$ matrix D is a second-order differencing matrix. The $(n-2)\times(n-2)$ matrix R^{-1} corresponds to the precision matrix of a MA(1) process (Brown and De Jong, 2001).

When covariates are unit spaced, we have the following expressions for D and R:

$$D = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & \vdots & & & \ddots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \end{bmatrix}, \; R = \begin{bmatrix} \frac{2}{3} & \frac{1}{6} & 0 & 0 & 0 & \cdots & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \cdots & 0 \\ & & & & & \ddots & \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{6} & \frac{2}{3} \end{bmatrix} \quad (5)$$

When covariates are unit spaced, we have the following expressions for D and R:

$$D = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 \\ \vdots & & & & \ddots & & \\ 0 & \vdots & & & & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} \frac{2}{3} & \frac{1}{6} & 0 & 0 & 0 & \cdots & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \cdots & 0 \\ & & & & & \ddots & \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{6} & \frac{2}{3} \end{bmatrix} \tag{5}$$

When covariates are unit spaced, we have the following expressions for D and R:

$$D = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & \vdots & & & \ddots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} \frac{2}{3} & \frac{1}{6} & 0 & 0 & 0 & \cdots & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \cdots & 0 \\ & & & & & \ddots & \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{6} & \frac{2}{3} \end{bmatrix} \tag{5}$$

- **Problem 1** : This ARIMA(0,2,1) interpretation requires covariate values to be equally spaced. Not directly applicable to irregular spaced locations.
- **Problem 2** : R^{-1} is a dense matrix, and hence the precision matrix of $\boldsymbol{g}$ is dense as well. Computation will be hard and not compatible with inference method such as Integrated Nested Laplace Approximation(INLA) (Rue et al., 2009).

Alternative: Extended Second Order Random Walk (RW2).

- **Problem 1** : This ARIMA(0,2,1) interpretation requires covariate values to be equally spaced. Not directly applicable to irregular spaced locations.
- **Problem 2** : R^{-1} is a dense matrix, and hence the precision matrix of $\boldsymbol{g}$ is dense as well. Computation will be hard and not compatible with inference method such as Integrated Nested Laplace Approximation(INLA) (Rue et al., 2009).

Alternative: Extended Second Order Random Walk (RW2).

- **Problem 1** : This ARIMA(0,2,1) interpretation requires covariate values to be equally spaced. Not directly applicable to irregular spaced locations.
- **Problem 2** : R^{-1} is a dense matrix, and hence the precision matrix of $\boldsymbol{g}$ is dense as well. Computation will be hard and not compatible with inference method such as Integrated Nested Laplace Approximation(INLA) (Rue et al., 2009).

Alternative: Extended Second Order Random Walk (RW2).

- **Problem 1** : This ARIMA(0,2,1) interpretation requires covariate values to be equally spaced. Not directly applicable to irregular spaced locations.
- **Problem 2** : R^{-1} is a dense matrix, and hence the precision matrix of $\boldsymbol{g}$ is dense as well. Computation will be hard and not compatible with inference method such as Integrated Nested Laplace Approximation(INLA) (Rue et al., 2009).

Alternative: Extended Second Order Random Walk (RW2).

From result of Wahba (1978), there is a well known connection between smoothing spline and folded Wiener process prior:

1. Let $W(t)$ denote the standard Wiener's process (Brownian motion), a SDE based prior is assigned to $g(t)$ in the following way ($\sigma_s = 1/\sqrt{\lambda}$):

$$\frac{d^2 g(t)}{dt^2} = \sigma_s \frac{dW(t)}{dt}.$$

2. The derivative of $W(t)$ does not exist in ordinary definition, but can be defined as a generalized function, the *white noise* process.
3. If $g(0)$ and $g'(0)$ are given diffuse Gaussian prior, the limiting posterior mean of $\boldsymbol{g}$ will be the minimizer of the smoothing spline problem (Wahba, 1978).

From result of Wahba (1978), there is a well known connection between smoothing spline and folded Wiener process prior:

1. Let $W(t)$ denote the standard Wiener's process (Brownian motion), a SDE based prior is assigned to $g(t)$ in the following way ($\sigma_s = 1/\sqrt{\lambda}$):

$$\frac{d^2 g(t)}{dt^2} = \sigma_s \frac{dW(t)}{dt}.$$

2. The derivative of $W(t)$ does not exist in ordinary definition, but can be defined as a generalized function, the *white noise* process.
3. If $g(0)$ and $g'(0)$ are given diffuse Gaussian prior, the limiting posterior mean of $\mathbf{g}$ will be the minimizer of the smoothing spline problem (Wahba, 1978).

From result of Wahba (1978), there is a well known connection between smoothing spline and folded Wiener process prior:

1. Let $W(t)$ denote the standard Wiener's process (Brownian motion), a SDE based prior is assigned to $g(t)$ in the following way ($\sigma_s = 1/\sqrt{\lambda}$):

$$\frac{d^2 g(t)}{dt^2} = \sigma_s \frac{dW(t)}{dt}.$$

2. The derivative of $W(t)$ does not exist in ordinary definition, but can be defined as a generalized function, the *white noise* process.
3. If $g(0)$ and $g'(0)$ are given diffuse Gaussian prior, the limiting posterior mean of $\boldsymbol{g}$ will be the minimizer of the smoothing spline problem (Wahba, 1978).

From result of Wahba (1978), there is a well known connection between smoothing spline and folded Wiener process prior:

1. Let $W(t)$ denote the standard Wiener's process (Brownian motion), a SDE based prior is assigned to $g(t)$ in the following way ($\sigma_s = 1/\sqrt{\lambda}$):

$$\frac{d^2 g(t)}{dt^2} = \sigma_s \frac{dW(t)}{dt}.$$

2. The derivative of $W(t)$ does not exist in ordinary definition, but can be defined as a generalized function, the *white noise* process.
3. If $g(0)$ and $g'(0)$ are given diffuse Gaussian prior, the limiting posterior mean of $\boldsymbol{g}$ will be the minimizer of the smoothing spline problem (Wahba, 1978).

The extended RW2 method of Lindgren and Rue (2008) can be derived from the following procedures:

1. Assign the SDE based prior on g.
2. Discretize the SDE into a finite dimensional problem using finite element method. Resulting in precision matrix being:

$$H^T B^{-1} H.$$

3. Apply a diagonal approximation to the tri-diagonal matrix B. Resulting in precision matrix being:

$$H^T A^{-1} H.$$

More technical details can be found in the report.

The extended RW2 method of Lindgren and Rue (2008) can be derived from the following procedures:

1. Assign the SDE based prior on g.
2. Discretize the SDE into a finite dimensional problem using finite element method. Resulting in precision matrix being:

$$H^T B^{-1} H.$$

3. Apply a diagonal approximation to the tri-diagonal matrix B. Resulting in precision matrix being:

$$H^T A^{-1} H.$$

More technical details can be found in the report.

The extended RW2 method of Lindgren and Rue (2008) can be derived from the following procedures:

1. Assign the SDE based prior on g.
2. Discretize the SDE into a finite dimensional problem using finite element method. Resulting in precision matrix being:

$$H^T B^{-1} H.$$

3. Apply a diagonal approximation to the tri-diagonal matrix B. Resulting in precision matrix being:

$$H^T A^{-1} H.$$

More technical details can be found in the report.

The extended RW2 method of Lindgren and Rue (2008) can be derived from the following procedures:

1. Assign the SDE based prior on g.
2. Discretize the SDE into a finite dimensional problem using finite element method. Resulting in precision matrix being:

$$H^T B^{-1} H.$$

3. Apply a diagonal approximation to the tri-diagonal matrix B. Resulting in precision matrix being:

$$H^T A^{-1} H.$$

More technical details can be found in the report.

The extended RW2 method of Lindgren and Rue (2008) can be derived from the following procedures:

1. Assign the SDE based prior on g.
2. Discretize the SDE into a finite dimensional problem using finite element method. Resulting in precision matrix being:

$$H^T B^{-1} H.$$

3. Apply a diagonal approximation to the tri-diagonal matrix B. Resulting in precision matrix being:

$$H^T A^{-1} H.$$

More technical details can be found in the report.

When covariates are unit spaced, we have the following expressions for H, B and A:

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 \\ & \vdots & & & \ddots & & \\ 0 & \vdots & & & \ddots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{1}{3} & \frac{1}{6} & 0 & 0 & 0 & \cdots & 0 & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \cdots & 0 & 0 \\ & & & & \ddots & & & \\ 0 & 0 & 0 & \cdots & \frac{1}{6} & \frac{2}{3} & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{6} & \frac{2}{3} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \frac{1}{6} & \frac{1}{3} \end{bmatrix}, \tag{6}$$

$$A = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ & & & & \ddots & & & \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}. \tag{7}$$

When covariates are unit spaced, we have the following expressions for H, B and A:

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & \vdots & & & \ddots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{1}{3} & \frac{1}{6} & 0 & 0 & 0 & \cdots & 0 & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \cdots & 0 & 0 \\ & & & & \ddots & & & \\ 0 & 0 & 0 & \cdots & \frac{1}{6} & \frac{2}{3} & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{6} & \frac{2}{3} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \frac{1}{6} & \frac{1}{3} \end{bmatrix}, \tag{6}$$

$$A = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ & & & & \ddots & & & \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}. \tag{7}$$

When covariates are unit spaced, we have the following expressions for H, B and A:

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & \vdots & & & \ddots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{1}{3} & \frac{1}{6} & 0 & 0 & 0 & \cdots & 0 & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \cdots & 0 & 0 \\ & & & & \ddots & & & \\ 0 & 0 & 0 & \cdots & \frac{1}{6} & \frac{2}{3} & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{6} & \frac{2}{3} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \frac{1}{6} & \frac{1}{3} \end{bmatrix}, \tag{6}$$

$$A = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ & & & & \ddots & & & \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}. \tag{7}$$

When covariates are unit spaced, we have the following expressions for H, B and A:

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & \vdots & & & \ddots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{1}{3} & \frac{1}{6} & 0 & 0 & 0 & \cdots & 0 & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \cdots & 0 & 0 \\ & & & & \ddots & & & \\ 0 & 0 & 0 & \cdots & \frac{1}{6} & \frac{2}{3} & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{6} & \frac{2}{3} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \frac{1}{6} & \frac{1}{3} \end{bmatrix}, \tag{6}$$

$$A = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ & & & & \ddots & & & \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}. \tag{7}$$

When covariates are unit spaced, we have the following expressions for H, B and A:

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & \vdots & & & \ddots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{1}{3} & \frac{1}{6} & 0 & 0 & 0 & \cdots & 0 & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \cdots & 0 & 0 \\ & & & & \ddots & & & \\ 0 & 0 & 0 & \cdots & \frac{1}{6} & \frac{2}{3} & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{6} & \frac{2}{3} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \frac{1}{6} & \frac{1}{3} \end{bmatrix}, \tag{6}$$

$$A = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ & & & & \ddots & & & \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}. \tag{7}$$

Lindgren and Rue (2008) implemented the extended RW2 method using the software INLA, which utilizes the sparseness of precision matrix to achieve efficient computation.

However, INLA software does not accommodate prior with dense precision matrix such as ARIMA(0,2,1).

Lindgren and Rue (2008) implemented the extended RW2 method using the software INLA, which utilizes the sparseness of precision matrix to achieve efficient computation.

However, INLA software does not accommodate prior with dense precision matrix such as ARIMA(0,2,1).

- Re-parametrizing the smoothing parameter σ_s^2 as $\theta = -2\log\sigma_s$, and let Q_θ denotes the precision matrix corresponding to the evaluation vector $\boldsymbol{g}$.
- Gaussian approximation:

$$\tilde{\pi}_G(\boldsymbol{g}|\boldsymbol{y},\theta) \propto \exp\left\{-\frac{1}{2}\left(\boldsymbol{g}-\hat{\boldsymbol{g}}_\theta\right)^T H_\theta(\hat{\boldsymbol{g}}_\theta)\left(\boldsymbol{g}-\hat{\boldsymbol{g}}_\theta\right)\right\}, \quad (8)$$

the quantity $\hat{\boldsymbol{g}}_\theta$ denotes $\text{argmax}_g \log\pi(\boldsymbol{g}|\theta,\boldsymbol{y})$ and $H_\theta(\boldsymbol{g})$ denotes $-\frac{d^2}{d\boldsymbol{g}d\boldsymbol{g}^T}\log\pi(\boldsymbol{g}|\theta,\boldsymbol{y})$.

- Re-parametrizing the smoothing parameter σ_s^2 as $\theta = -2\log\sigma_s$, and let Q_θ denotes the precision matrix corresponding to the evaluation vector $\boldsymbol{g}$.
- Gaussian approximation:

$$\tilde{\pi}_G(\boldsymbol{g}|\boldsymbol{y},\theta) \propto \exp\left\{-\frac{1}{2}\left(\boldsymbol{g}-\hat{\boldsymbol{g}}_\theta\right)^T H_\theta(\hat{\boldsymbol{g}}_\theta)\left(\boldsymbol{g}-\hat{\boldsymbol{g}}_\theta\right)\right\}, \quad (8)$$

the quantity $\hat{\boldsymbol{g}}_\theta$ denotes $\text{argmax}_g \log\pi(\boldsymbol{g}|\theta,\boldsymbol{y})$ and $H_\theta(\boldsymbol{g})$ denotes $-\frac{d^2}{dgdg^T}\log\pi(\boldsymbol{g}|\theta,\boldsymbol{y})$.

- Re-parametrizing the smoothing parameter σ_s^2 as $\theta = -2\log\sigma_s$, and let Q_θ denotes the precision matrix corresponding to the evaluation vector $\boldsymbol{g}$.
- Gaussian approximation:

$$\tilde{\pi}_G(\boldsymbol{g}|\boldsymbol{y},\theta) \propto \exp\left\{-\frac{1}{2}\left(\boldsymbol{g}-\hat{\boldsymbol{g}}_\theta\right)^T H_\theta(\hat{\boldsymbol{g}}_\theta)\left(\boldsymbol{g}-\hat{\boldsymbol{g}}_\theta\right)\right\}, \quad (8)$$

the quantity $\hat{\boldsymbol{g}}_\theta$ denotes $\text{argmax}_g \log\pi(\boldsymbol{g}|\theta,\boldsymbol{y})$ and $H_\theta(\boldsymbol{g})$ denotes $-\frac{d^2}{d\boldsymbol{g}d\boldsymbol{g}^T}\log\pi(\boldsymbol{g}|\theta,\boldsymbol{y})$.

- Obtain the Laplace approximation as Tierney and Kadane (1986):

$$\tilde{\pi}_{\mathrm{LA}}(\theta|\boldsymbol{y}) \propto \pi(\theta)\left\{\frac{|Q_\theta|}{|H_\theta(\hat{\boldsymbol{g}}_\theta)|}\right\}^{1/2} \exp\left\{-\frac{1}{2}\hat{\boldsymbol{g}}_\theta^T Q_\theta \hat{\boldsymbol{g}}_\theta + l(\boldsymbol{y};\hat{\boldsymbol{g}}_\theta)\right\}. \tag{9}$$

- Numerical Integration:

$$\tilde{\pi}(\boldsymbol{g}|\boldsymbol{y}) = \sum_{k=1}^{K} \tilde{\pi}_G(\boldsymbol{g}|\boldsymbol{y},\theta_k)\tilde{\pi}_{\mathrm{LA}}(\theta_k|\boldsymbol{y})\delta_k, \tag{10}$$

  where $\{\theta_k,\delta_k\}_{k=1}^K$ is a set of K nodes and weights selected using Adaptive Gauss-Hermite Quadrature (AGHQ) rule (Stringer, 2021).
- The computation of the AGHQ rule requires optimization of $\tilde{\pi}_{\mathrm{LA}}(\theta|\boldsymbol{y})$, which will be done through the TMB package (Kristensen et al., 2015) with automatic differentiation.

- Obtain the Laplace approximation as Tierney and Kadane (1986):

$$\tilde{\pi}_{\mathrm{LA}}(\theta|\boldsymbol{y}) \propto \pi(\theta)\left\{\frac{|Q_\theta|}{|H_\theta(\hat{\boldsymbol{g}}_\theta)|}\right\}^{1/2} \exp\left\{-\frac{1}{2}\hat{\boldsymbol{g}}_\theta^T Q_\theta \hat{\boldsymbol{g}}_\theta + l(\boldsymbol{y};\hat{\boldsymbol{g}}_\theta)\right\}. \tag{9}$$

- Numerical Integration:

$$\tilde{\pi}(\boldsymbol{g}|\boldsymbol{y}) = \sum_{k=1}^{K} \tilde{\pi}_G(\boldsymbol{g}|\boldsymbol{y},\theta_k)\tilde{\pi}_{\mathrm{LA}}(\theta_k|\boldsymbol{y})\delta_k, \tag{10}$$

where $\{\theta_k, \delta_k\}_{k=1}^K$ is a set of K nodes and weights selected using Adaptive Gauss-Hermite Quadrature (AGHQ) rule (Stringer, 2021).

- The computation of the AGHQ rule requires optimization of $\tilde{\pi}_{\mathrm{LA}}(\theta|\boldsymbol{y})$, which will be done through the TMB package (Kristensen et al., 2015) with automatic differentiation.

- Obtain the Laplace approximation as Tierney and Kadane (1986):

$$\tilde{\pi}_{\text{LA}}(\theta|\boldsymbol{y}) \propto \pi(\theta)\left\{\frac{|Q_\theta|}{|H_\theta(\hat{\boldsymbol{g}}_\theta)|}\right\}^{1/2} \exp\left\{-\frac{1}{2}\hat{\boldsymbol{g}}_\theta^T Q_\theta \hat{\boldsymbol{g}}_\theta + l(\boldsymbol{y};\hat{\boldsymbol{g}}_\theta)\right\}. \tag{9}$$

- Numerical Integration:

$$\tilde{\pi}(\boldsymbol{g}|\boldsymbol{y}) = \sum_{k=1}^{K} \tilde{\pi}_G(\boldsymbol{g}|\boldsymbol{y},\theta_k)\tilde{\pi}_{\text{LA}}(\theta_k|\boldsymbol{y})\delta_k, \tag{10}$$

where $\{\theta_k, \delta_k\}_{k=1}^K$ is a set of K nodes and weights selected using Adaptive Gauss-Hermite Quadrature (AGHQ) rule (Stringer, 2021).

- The computation of the AGHQ rule requires optimization of $\tilde{\pi}_{\text{LA}}(\theta|\boldsymbol{y})$, which will be done through the TMB package (Kristensen et al., 2015) with automatic differentiation.

- Obtain the Laplace approximation as Tierney and Kadane (1986):

$$\tilde{\pi}_{\text{LA}}(\theta|\boldsymbol{y}) \propto \pi(\theta)\left\{\frac{|Q_\theta|}{|H_\theta(\hat{\boldsymbol{g}}_\theta)|}\right\}^{1/2} \exp\left\{-\frac{1}{2}\hat{\boldsymbol{g}}_\theta^T Q_\theta \hat{\boldsymbol{g}}_\theta + l(\boldsymbol{y};\hat{\boldsymbol{g}}_\theta)\right\}. \tag{9}$$

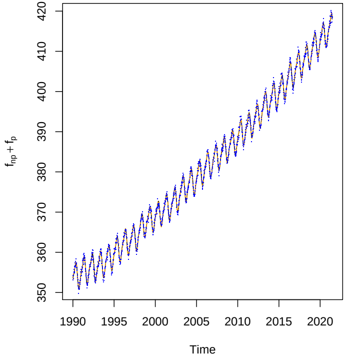
- Numerical Integration:

$$\tilde{\pi}(\boldsymbol{g}|\boldsymbol{y}) = \sum_{k=1}^{K} \tilde{\pi}_G(\boldsymbol{g}|\boldsymbol{y},\theta_k)\tilde{\pi}_{\text{LA}}(\theta_k|\boldsymbol{y})\delta_k, \tag{10}$$

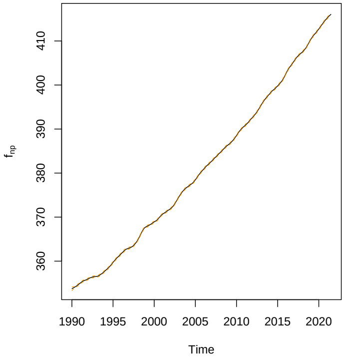
where $\{\theta_k, \delta_k\}_{k=1}^K$ is a set of K nodes and weights selected using Adaptive Gauss-Hermite Quadrature (AGHQ) rule (Stringer, 2021).

- The computation of the AGHQ rule requires optimization of $\tilde{\pi}_{\text{LA}}(\theta|\boldsymbol{y})$, which will be done through the TMB package (Kristensen et al., 2015) with automatic differentiation.

Going back to the CO2 Example with RW2 method:



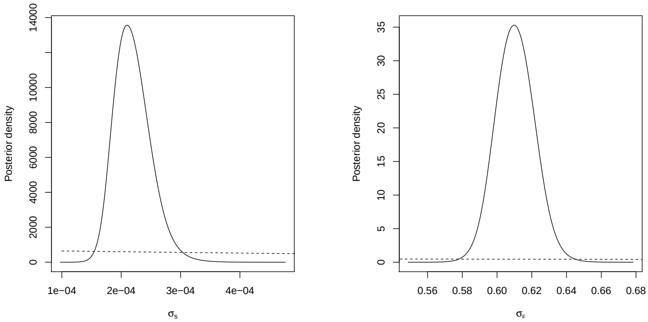
(a) Overall effect f



(b) Random effect f_{np}

Figure: Inference obtained for the CO2 dataset using RW2, for observations after year 1990.

Looking at the variance and smoothing parameters:

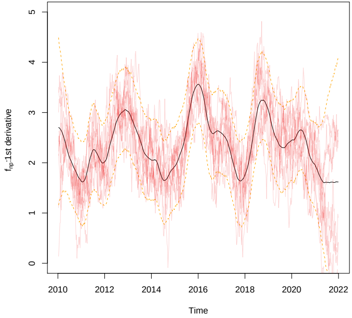


(a) Posterior for the smoothing parameter σ_s

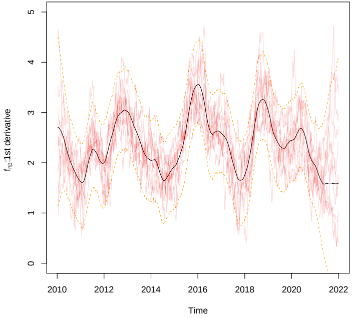
(b) Posterior for the variance parameter σ_ϵ

Figure: Inference for smoothing/variance parameters with PC prior (Simpson et al., 2017) with median 2.

What about the derivatives?

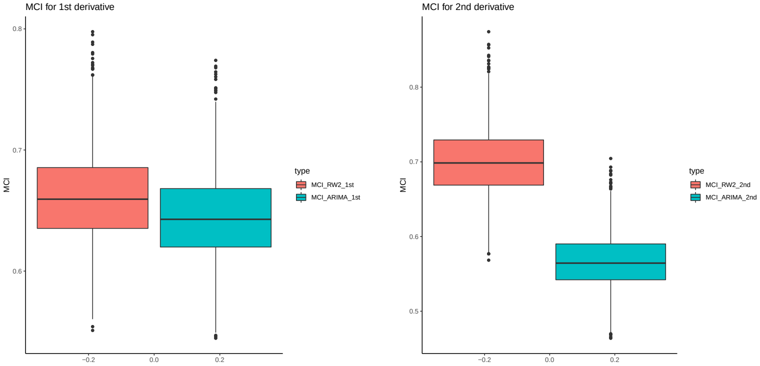


(a) f'_{np} using RW2



(b) f'_{np} using ARIMA

A Simulation Study with 1000 independent replications:



(c) MCI for g' (d) MCI for g''

Figure: Mean width of 90 percent credible interval (MCI) for g', g'' using RW2 or ARIMA, replicated for 1000 independent data sets.

- We provide an overview of the extended second order random walk method (Lindgren and Rue, 2008), as well as its connection with the smoothing spline (Wahba, 1978) and the ARIMA prior (Brown and De Jong, 2001).
- The RW2 method gives similar result in terms of inference for g as the ARIMA method, but less smooth inference for higher order derivatives of g compared to ARIMA method.
- We illustrate that It is possible to implement the exact ARIMA method with dense precision matrix. But which method is better should depend on the question of interest.

- We provide an overview of the extended second order random walk method (Lindgren and Rue, 2008), as well as its connection with the smoothing spline (Wahba, 1978) and the ARIMA prior (Brown and De Jong, 2001).
- The RW2 method gives similar result in terms of inference for g as the ARIMA method, but less smooth inference for higher order derivatives of g compared to ARIMA method.
- We illustrate that It is possible to implement the exact ARIMA method with dense precision matrix. But which method is better should depend on the question of interest.

- We provide an overview of the extended second order random walk method (Lindgren and Rue, 2008), as well as its connection with the smoothing spline (Wahba, 1978) and the ARIMA prior (Brown and De Jong, 2001).
- The RW2 method gives similar result in terms of inference for g as the ARIMA method, but less smooth inference for higher order derivatives of g compared to ARIMA method.
- We illustrate that It is possible to implement the exact ARIMA method with dense precision matrix. But which method is better should depend on the question of interest.

- We provide an overview of the extended second order random walk method (Lindgren and Rue, 2008), as well as its connection with the smoothing spline (Wahba, 1978) and the ARIMA prior (Brown and De Jong, 2001).
- The RW2 method gives similar result in terms of inference for g as the ARIMA method, but less smooth inference for higher order derivatives of g compared to ARIMA method.
- We illustrate that It is possible to implement the exact ARIMA method with dense precision matrix. But which method is better should depend on the question of interest.

Brown, P. and P. De Jong (2001, March). Nonparametric smoothing using state space techniques. *Canadian Journal of Statistics 29*(1), 37–50.

Kristensen, K., A. Nielsen, C. W. Berg, H. Skaug, and B. Bell (2015). Tmb: automatic differentiation and laplace approximation. *arXiv preprint arXiv:1509.00660*.

Lindgren, F. and H. Rue (2008). On the second-order random walk model for irregular locations. *Scandinavian Journal of Statistics 35*(4), 691–700.

Rue, H., S. Martino, and N. Chopin (2009). Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 71*(2), 319–392.

Simpson, D., H. Rue, A. Riebler, T. G. Martins, and S. H. Sørbye (2017). Penalising model component complexity: A principled, practical approach to constructing priors. *Statistical science 32*(1), 1–28.

Stringer, A. (2021). Implementing approximate bayesian inference using adaptive quadrature: the aghq package.

Tierney, L. and J. B. Kadane (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the american statistical association 81*(393), 82–86.

Wahba, G. (1978). Improper priors, spline smoothing and the problem of guarding against model errors in regression. *Journal of the Royal Statistical Society. Series B (Methodological) 40*(3), 364–372.