Screening of airborne radioxenon measurements for Nuclear-Test-Ban Treaty verification

Jerome Alhage, Jelle Bosmans, Thomas Ceulemans, Nick Dewaele

Abstract

We extend a simulation model for radioxenon emissions by inferring daily emissions of sources in the Northern Hemisphere. This is done by numerical optimisation of the relative prediction error over the daily emissions.

1 Introduction

In this paper, we adapt a simulation model of radioxenon emissions used at SCK-CEN for the verification of the Nuclear-Test-Ban Treaty. This model operates on a map of $m = 200$ civilian sources of radioxenon with their average daily emissions $x_1, \ldots, x_m$. A list of $n = 4636$ measurements $[y_1, \ldots, y_n]$ at stations spread across the Northern Hemisphere is given. The relationship between the emissions and the measured concentrations is given by a linear model

$$y_i \approx \hat{y}_i = \sum_{j=1}^{m} \sum_{\Delta=1}^{15} M_{i,(j,\Delta)} \frac{x_j}{s_j} \tag{1}$$

where the sensitivities $M_{i,(j,\Delta)}$ are obtained from backward simulation of the so-called *xenon weather*. The scaling factors s_j arise from the simulation on a mesh of the Northern Hemisphere. The *sensitivity matrices* $M_{i,(j,\Delta)}$ can be interpreted as the contribution of the emission from source j from Δ days before the measurement to the i'th concentration measurement in the dataset. The contributions of more than 15 days before the measurement are neglected.

1.1 Outline

The above assumes that the xenon emission is constant for every source. We adapt this model in section 2 so that daily emissions are taken into account. In section 3, we discuss the use of a transformation from [1] to model the accuracy in a non-detection event. In section 4, we explain a method to fit the extended model to the data. Given the time constraints of this project, the optimality criterion was deliberately kept simple. Numerical data in section 5 show that the average prediction error can be increased. We conclude in section 6 with suggestions make the model more realistic and improve computational aspects.

1.2 Notation

- $\mathbf{1}_q = [1 \ \ldots \ 1]^T \in \mathbb{R}^q$ is the constant vector consisting of only ones.
- $\odot$ is the Hadamard (i.e. elementwise) product, given by $(AB)_{ij} = A_{ij}B_{ij}$.
- vec : $\mathbb{R}^{p\times q} \to \mathbb{R}^{pq}$ is the vectorisation operator. It can be understood as stacking the columns of a matrix in one long vector.
- diag is the canonical identification of p-vectors and $p \times p$ diagonal matrices, i.e.,

$$\operatorname{diag}\left(\begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix}\right) = \begin{bmatrix} a_1 & 0 & \ldots & 0 \\ 0 & a_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & a_p \end{bmatrix}.$$

- $\otimes$ is the Kronecker product. In particular,

$$\underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix}}_{\in\mathbb{R}^p} \otimes \underbrace{\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}}_{\in\mathbb{R}^q} = \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_1 \\ a_2 \\ \vdots \\ a_2 \\ \vdots \\ a_p \end{bmatrix}}_{\in\mathbb{R}^{pq}}.$$

  A standard rule of thumb in numerical linear algebra is that the Kronecker product serves a mostly theoretical purpose and is rarely calculated explicitly. This is because the Kronecker product satisfies $\mathrm{vec}(AXB) = (B^T \otimes A)\mathrm{vec}X$ for all matrices A, X, B [2, eq. 1.3.6].

- For a vector-valued function $F(x)$, the Jacobian matrix is denoted as $\frac{\partial F}{\partial x}$.

2 Model

2.1 Reformulation of the baseline model

Equation (1) can be formulated as follows:

$$\begin{aligned} \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix} &= \begin{bmatrix} M_{1,(1,1)} & M_{1,(1,2)} & \cdots & M_{1,(m,15)} \\ \vdots & \vdots & & \vdots \\ M_{n,(1,1)} & M_{n,(1,2)} & \cdots & M_{n,(m,15)} \end{bmatrix} \begin{bmatrix} s_1^{-1}x_1 \\ \vdots \\ s_1^{-1}x_1 \\ s_2^{-1}x_2 \\ \vdots \\ s_n^{-1}x_m \end{bmatrix} \\ &= \begin{bmatrix} M_{1,(1,1)} & M_{1,(1,2)} & \cdots & M_{1,(n,15)} \\ \vdots & \vdots & & \vdots \\ M_{m,(1,1)} & M_{m,(1,2)} & \cdots & M_{m,(n,15)} \end{bmatrix} \left(S^{-1} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \otimes \mathbf{1}_{15} \right) \end{aligned} \tag{2}$$

where $S = \mathrm{diag}([s_1, \ldots, s_m])$. In this section, we present a different interpretation. Instead of $S^{-1}x \otimes \mathbf{1}_{15}$, we make a vector $\tilde{x} := S^{-1}x \otimes \mathbf{1}_T$ where $T = 365 + 15 = 380$. This is a vector of length mT that gives the emission from the most recent day to T days in the past for each source. This is assumed to be constant for all days. Therefore, $\tilde{x}$ consists of $m = 200$ blocks, each of which is a constant vector of length T.

We can write a system of equations that is logically the same as eq. (2) but uses $\tilde{x}$ instead of $S^{-1}x \otimes \mathbf{1}_{15}$. This gives a larger linear system. Because the emission from most dates has a negligible contribution to the observed concentrations, many coefficients in the larger matrix will be zero. For every measurement i, we have the linear relationship

$$\hat{y}_i = \begin{bmatrix} \cdots & M_{i,(1,1)} & \ldots & M_{i,(1,15)} & 0 & \ldots & M_{i,(2,1)} & \ldots & M_{i,(2,15)} & \ldots \end{bmatrix} \tilde{x}.$$

The presence of zeros in the matrix indicates that emissions from the distant past do not contribute to the observed concentrations.

Assume for the sake of simplicity that in each station, there is one observation per day, indexed as $y_{k,t}$, where $k = 1, \ldots, K$ where K is the number of stations and $t = 1, \ldots, 365$. The index $t = 1$ corresponds to the most recent measurement. Then the full system reads as follows:

$$\begin{bmatrix} \hat{y}_{1,1} \\ \hat{y}_{1,2} \\ \vdots \\ \hat{y}_{2,1} \\ \vdots \\ \hat{y}_{K,365} \end{bmatrix} = \underbrace{\begin{bmatrix} M_{(1,1),(1,1)} & M_{(1,1),(1,2)} & \cdots & M_{(1,1),(1,15)} & 0 & \cdots & \cdots & M_{(1,1),(2,1)} & \cdots & 0 \\ 0 & M_{(1,2),(1,1)} & \cdots & M_{(1,2),(1,14)} & M_{(1,2),(1,15)} & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & \vdots \\ M_{(2,1),(1,1)} & M_{(2,1),(1,2)} & \cdots & M_{(2,1),(1,15)} & 0 & \cdots & \cdots & M_{(2,1),(2,1)} & \cdots & 0 \\ \vdots & & & & & & & & & \\ \vdots & & & & & & & & & \end{bmatrix}}_{:=M_{shift}} \tilde{x}. \tag{3}$$

Remark 1. *As an alternative to using $\tilde{x} = S^{-1}x \otimes \mathbf{1}_T$, one could use the permuted expression $\mathbf{1}_T \otimes (S^{-1}x)$. This is a vector containing all emissions on day 1, followed by all emissions on day 2, etc. A similar permutation of y could be chosen. These permutations can be compensated by the inverse permutation on the columns and rows of M_{shift}, respectively. The resulting model is essentially identical to eq.* (3) *but may have advantages when it comes to implementation in software. We will not consider this.*

2.2 Model with time-dependent correction to the emissions

Daily emissions can be modelled by multiplying the average emission of source j by some scaling factor $w_{j,t}$. This corresponds to the transformation

$$\begin{bmatrix} s_1^{-1}x_1 \\ \vdots \\ s_1^{-1}x_1 \\ s_2^{-1}x_2 \\ \vdots \\ s_m^{-1}x_m \end{bmatrix} \mapsto \begin{bmatrix} w_{1,1}s_1^{-1}x_1 \\ \vdots \\ w_{1,T}s_1^{-1}x_1 \\ w_{2,1}s_2^{-1}x_2 \\ \vdots \\ w_{m,T}s_m^{-1}x_m \end{bmatrix}$$

which can be written concisely as $W\tilde{x}$ where $W = \operatorname{diag}([w_{1,1}, w_{1,2}, \ldots, w_{m,T}])$. Using this weighted vector of emissions instead of $\tilde{x}$ in eq. (3) gives

$$\hat{y} = M_{shift}W\tilde{x}. \tag{4}$$

In the next section, we present a method to fit the unknown factors W to the data.

3 Correction for non-detections

Sometimes, the concentration at a measurement station is below the minimum detectable concentration (MDC). The corresponding measurements are $y_i = 0$. To account for non-detections, we apply the transformation h from [1]. This is given by the continuously differentiable function

$$h(y) = \begin{cases} \frac{y^2}{4MDC} + MDC & \text{if } y \leq 2MDC \\ y & \text{else} \end{cases}.$$

This transformation maps the interval $[0, MDC]$ onto $[MDC, 2MDC]$. For observations that are significantly larger than MDC, this is approximately the identity. If y and the simulated value $\hat{y}$ are both larger than $2MDC$, then $h(\hat{y})/h(y) = \hat{y}/y$. If $y = 0$ because the concentration is not detectable, then we want a simulated value $\hat{y}$ such that $h(\hat{y}) \approx MDC$. In both cases, we can use the value of $h(\hat{y})/h(y)$ as an indication of the accuracy of the simulation.

In the dataset that we are working with, we know the values of $-\frac{M_{i,(j,\Delta)}}{MDC}$ for every non-detection event. In these events, we work as if $MDC = 1$. If the predicted value is $\hat{y}_i < 0$, we have

$$h(\hat{y}) = \begin{cases} \frac{\hat{y}^2}{4} + 1 & \text{if } \hat{y} \leq -2 \\ -\hat{y} & \text{else} \end{cases}$$

and $h(y) = 1$. For the detection events, we do not have the MDC and cannot calculate h exactly. For the detection events, we can approximate h by the identity map.

4 Fitting the model

In the above, we argued that the prediction is accurate if $h(\hat{y})$ is close to one, where $\hat{y}$ is the predicted concentration. This is equivalent to $\log^2(h(\hat{y}))$ being small. Notice that this penalises overestimates by a factor α as much as underestimates by a factor α. That is,

$$\log^2(\alpha) = \log^2(\alpha^{-1}).$$

The average prediction error over all predictions is $\sum_{i=1}^{n} \log^2(h(\hat{y}_i)) = \|\log h.(\hat{y})\|^2$, where $\log h.$ is the elementwise application of $x \mapsto \log h(x)$. This will be the first criterion for optimality of the model.

The second criterion is the ratio between the daily emissions and the average emission for each source, i.e. the factors W. In a similar vein as the above, the size of W can be measured as $\|\log W\|^2$ where $\|\cdot\|$ is the Frobenius norm[1].

[1] Since W is a diagonal matrix, the matrix logarithm $\log W$ is simply the diagonal matrix with $\log w_{1,1}, \log w_{1,2}, \ldots$ on the diagonal. Hence, $\|\log W\|^2 = \log^2 w_{1,1} + \log^2 w_{1,2} + \ldots$

To balance these two criteria, we measure the quality of the model as

$$f(W) := \underbrace{\frac{1}{2}\|\log h.(\hat{y}(W))\|^2}_{\text{prediction error}} + \underbrace{\frac{1}{2}\lambda\|\log W\|^2}_{\text{deviation from average emission}} \quad \text{where} \quad \hat{y}(W) = M_{shift} W \tilde{x} \tag{5}$$

where $\lambda \geq 0$ is a regularisation parameter.

Observe that the model where W is the identity matrix $\mathbb{I}$ is equivalent to the original model. Furthermore, the regularisation term vanishes if and only if $W = \mathbb{I}$. That is,

$$f(\mathbb{I}) = \frac{1}{2}\|\log h.(\hat{y}(\mathbb{I}))\|^2$$

is the prediction error for the original model. For $W \neq \mathbb{I}$, the regularisation term is strictly positive. This means that, if $f(W) \leq f(\mathbb{I})$, then the prediction error is decreased, i.e.,

$$\frac{1}{2}\|\log h.(\hat{y}(W))\|^2 \leq \frac{1}{2}\|\log h.(\hat{y}(\mathbb{I}))\|^2.$$

Hence, the model corresponding to the minimum of eq. (5) predicts the data at least as accurately as the original model.

To optimise f numerically, we will parametrise W in eq. (5) as $W = e^{\operatorname{diag} v}$ where $v = [v_1, v_2, \dots]$ is some vector and $e^{\operatorname{diag} v}$ is the diagonal matrix with $[e^{v_1}, e^{v_2}, \dots]$ on the diagonal. The motivation for doing this is twofold:

1. It ensures that W remains strictly positive. In particular, the diagonal elements of W can only tend towards zero if $v_j \to -\infty$ for some i.
2. It implicitly measures changes to W in a relative sense. If the optimisation algorithm performs a linear update $v_{k+1} = v_k + \Delta v$, this corresponds to a relative update $W_{k+1} = W_k e^{\operatorname{diag} \Delta v}$.

Thus, in practice, we are interested in the optimisation problem

$$\min_v f(v) \quad \text{where} \quad f(v) = \frac{1}{2}\|\log h.(\hat{y})\|^2 + \frac{1}{2}\lambda\|v\|^2 \quad \text{and} \quad \hat{y} = M_{shift} e^{\operatorname{diag} v} \tilde{x}. \tag{6}$$

Most optimisation algorithms require at least the gradient of f. This is given by the following:

Proposition 2. *Let $g = [g_1, \dots, g_n]$ be the vector defined elementwise by*

$$g_i = \frac{\log h(\hat{y}_i)}{h(\hat{y}_i)} \frac{\mathrm{d}}{\mathrm{d}\hat{y}_i} h(\hat{y}_i).$$

Then

$$\operatorname{grad} f(v) = e^v \odot (M_{shift}^T g) \odot \tilde{x}$$

where e^v denotes elementwise exponentiation and $\odot$ is the Hadamard (elementwise) product.

Proof. Let $W = e^{\operatorname{diag} v}$ and let U be the matrix corresponding to the linear map $\operatorname{vec} \circ \operatorname{diag}$. Then $\operatorname{vec} W(v) = U e^v$. Define

$$F(v) := \log h.(\hat{y}(v)).$$

By using basic properties of the Kronecker product [2, eq. 1.3.6], we have

$$\hat{y}(v) = M_{shift} W(v) \tilde{x} = (\tilde{x}^T \otimes M_{shift}) \operatorname{vec} W(v) = (\tilde{x}^T \otimes M_{shift}) U e^v,$$

so that

$$\frac{\partial F}{\partial v} = \underbrace{\operatorname{diag}\left(\frac{\mathrm{d}}{\mathrm{d}x} \log h(x) \Big|_{x=\hat{y}(v)} \right)}_{:=H} \frac{\partial \hat{y}}{\partial v} = H(\tilde{x}^T \otimes M_{shift}) U e^{\operatorname{diag} v}.$$

The matrix H appearing in this expression is defined by the diagonal elements

$$H_{ii} = \frac{1}{h(\hat{y}_i)} \left(\frac{\mathrm{d}}{\mathrm{d}x} h(x) \Big|_{x=\hat{y}_i} \right).$$

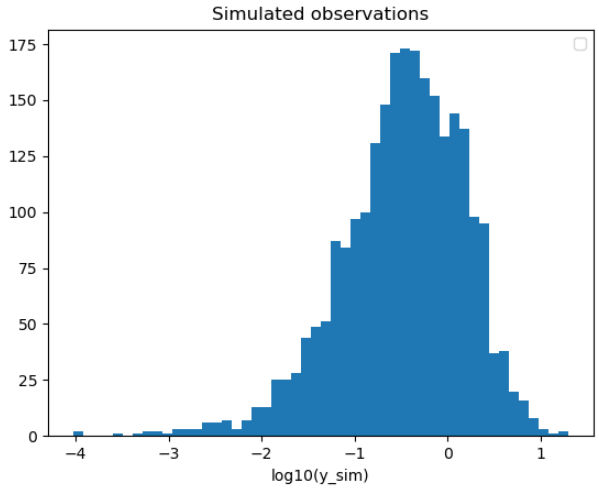

Figure 1: Distribution of the simulated observation using the yearly averaged emissions at all times.

Using standard differentiation rules for the squared norm of a vector-valued function, we obtain

$$\begin{aligned}\operatorname{grad}_v\left(\frac{1}{2}\|F(v)\|^2\right) &= \left(\frac{\partial F}{\partial v}\right)^T F(v)\\ &= e^{\operatorname{diag} v} U^T(\tilde{x}\otimes M_{shift}^T) H \log h.(\hat{y})\\ &= e^{\operatorname{diag} v} U^T(\tilde{x}\otimes M_{shift}^T) g\\ &= e^{\operatorname{diag} v} U^T \operatorname{vec}(M_{shift}^T g \tilde{x}^T)\end{aligned}$$

Since U is the unitary map that maps a vector onto (the vectorisation of) the corresponding diagonal matrix, U^T maps the vectorisation of a matrix onto its diagonal. Since the diagonal of the rank-1 matrix $M_{shift}^T g\tilde{x}^T$ is $(M_{shift}^T g)\odot\tilde{x}$, the above simplifies to

$$\operatorname{grad}_v\left(\frac{1}{2}\|F(v)\|^2\right) = e^{\operatorname{diag} v}\left((M_{shift}^T g)\odot\tilde{x}\right) = e^v\odot(M_{shift}^T g)\odot\tilde{x}.$$

The foregoing gives the derivative of the first term in the definition of $f(v)$. The gradient of the second term is simply λv. □

With this information, we can optimise eq. (6) using any gradient-based optimisation method. We decided on limited-memory BFGS [3], which is implimented in SciPy [4].

5 Numerical results

Before applying any optimization procedure, we must establish a baseline. This is done by filling in the daily source emissions x in eq. (1) with their yearly averages. The simulated observations in fig. 1 seem to be biased towards lower values than expected given the actual measurements.

In order to quantify the performance of the simulations, we define some metrics. First of all, the fraction of the simulated observations which lie within a factor 2 of the actual observations. In the ideal case, all simulated observations $\hat{y}$ would be exactly the same as the observed ones y. This metric allows for some small scaling factor as error.

$$\text{FAC2} := P(0.5 \le \hat{y}/y \le 2) \tag{7}$$

The previous metric only takes into account the simulated observations within a small area. Therefore it cannot tell us whether some general bias is present. To investigate this, we use the geometric mean bias. The more biased the ratio is, the further this value will lie from 1. This is because it defines some mean ratio of actual observations and simulated ones.

$$\text{GM} := e^{-E(ln(\hat{y}/y))} \tag{8}$$

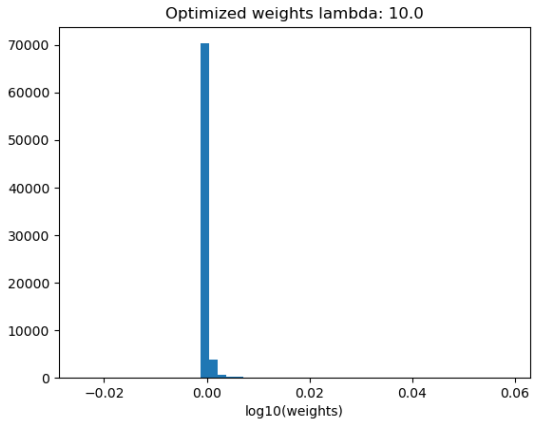

Figure 2: Histogram of the optimized weights W, using a large regularization parameter $\lambda = 10$.

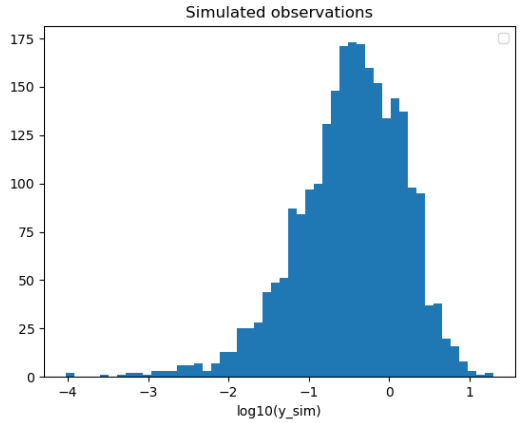

Figure 3: Histogram of the ratio of simulated observations $\hat{y}/y$, using a large regularization parameter $\lambda = 10$.

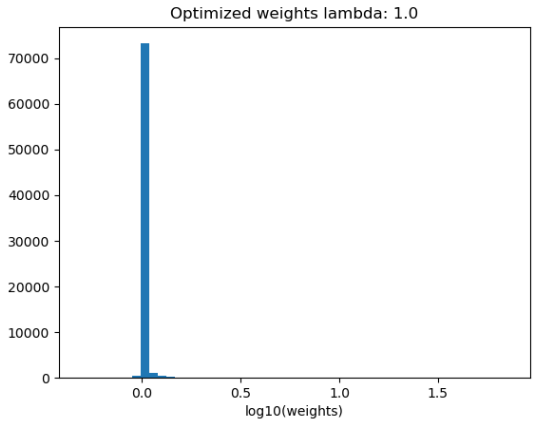

Figure 4: Histogram of the optimized weights W, using a medium regularization parameter $\lambda = 1$.

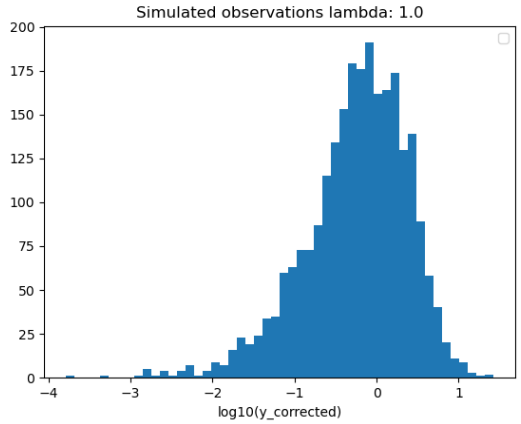

Figure 5: Histogram of the ratio of simulated observations $\hat{y}/y$, using a medium regularization parameter $\lambda = 1$.

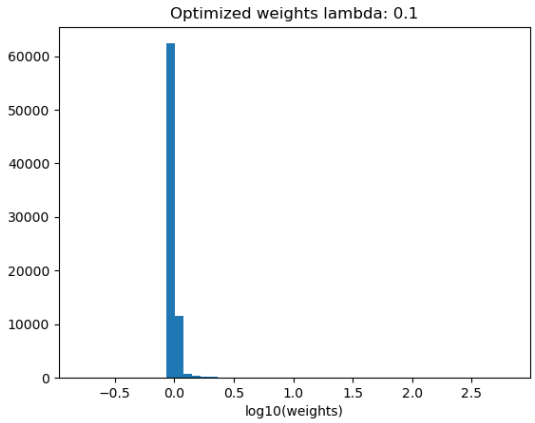

Figure 6: Histogram of the optimized weights W, using a small regularization parameter $\lambda = 0.1$.

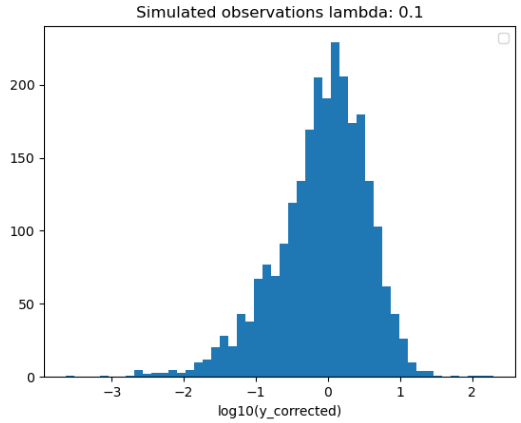

Figure 7: Histogram of the ratio of simulated observations $\hat{y}/y$, using a small regularization parameter $\lambda = 0.1$.

Table 1: Metrics applied to the simulated observations in case of different regularization parameters λ.

	FAC2	GM
baseline	31.7%	3.050
$\lambda = 10$	32.2%	2.984
$\lambda = 1$	40.5%	1.776
$\lambda = 0.1$	40.8%	1.216

Starting with the results when using a high regularization parameter $\lambda = 10$, we observe that the weights are nearly 1 everywhere (see fig. 2). This makes sense, as the cost function in this case heavily penalizes the weights for deviating from 1. As a result, the histogram of simulated results (see fig. 3) is nearly identical to the baseline. Lowering the regularization parameter $\lambda = 1$, we find that the weights have more freedom to deviate from 1 (see fig. 4) and the bias on the ratio $\hat{y}/y$ decreases (see fig. 5). This is a direct result of lowering the influence of the regularization term. Finally, lowering the regularization parameter $\lambda = 0.1$ even further, the weights get even more freedom (see fig. 6), while the ratio $\hat{y}/y$ finally seems to be centered around 1. In table 1, a summary of the metrics is found, evaluated at the different values for the regularization parameter. Summarized, we see the most major change in the geometric mean bias GM as the regularization parameter decreases. This might lead us to believe that the best result is obtained by choosing λ as small as possible. However, a small value for λ results in larger deviations from the mean emission; these can become greater than can what be expected in real life TODO REF PAPER typical variation. Thus a fine balance between the prediction error and the regularization condition is still needed.

6 Further work

- Standard software libraries support optimisation with constraints. This is an alternative to regularisation and may be more interpretable. Suppose that the number of components in $\tilde{x}$ is mT (where m is the number of sources and T is the number of days). If we want to bound the average of $(\log w_{ij})^2$ by some value δ^2, then we have the constrained optimisation problem

$$\begin{aligned} \min \quad & \frac{1}{2}\|\log h.(\hat{y})\|^2 \\ \text{s.t.} \quad & \frac{1}{mT}\|\log W\|^2 \leq \delta^2 \end{aligned}$$

  which can be solved using similar ideas as those developed above. The prototypical constrained optimisation method (such as an active set method) would iterate towards a critical point of the Lagrangian

$$\mathcal{L}(W, \lambda) = \frac{1}{2}\|\log h.(\hat{y})\|^2 + \lambda\left(\frac{1}{mT}\|\log W\|^2 - \delta^2\right)$$

  in which case the computational cost would be comparable to optimising eq. (5). A more detailed explanation of constrained optimisation and the Lagrangian is found in [3].

- Currently, there is no constraint saying that, for every emission source j, the geometric mean of the emission should be the reported average x_j. This can be enforced with a constraint like $\prod_t w_{jt} = 1$ for all sources $j = 1, \ldots, m$. With the parametrisation $w = e^v$, this is equivalent to the linear constraint $\sum_t v_{jt} = 0$. Since this is a linear constraint, it can be handled relatively easily.

- For certain emission sources in certain periods, there is an expected correlation between the emission at time t and at time $t + 1, t + 2, \ldots$. We can add a constraint on the autocorrelation.

- Time-correlation could be measured by parameterising the model as follows:

$$\begin{aligned} x_1^{2\ \text{Jan 2014}} &= \alpha_1^{(1)} x_1^{1\ \text{Jan 2014}} \\ x_1^{3\ \text{Jan2014}} &= \alpha_2^{(1)} x_1^{2\ \text{Jan 2014}} \\ &\vdots \end{aligned}$$

  with a constraint like $\alpha_{\min} \leq \prod_{t=1}^{T} \alpha_t^{(j)} \leq \alpha_{\max}$ for all T and all $j = 1, \ldots, m$. However, this would be difficult to implement. In addition, the complexity of inequality constrained optimisation can be combinatorial in the number of constraints if an active set method is used [3].

- For the sake of simplicity, the current implementation does not store M_{shift} in memory in a data structure for large and sparse matrices. This could be changed to improve memory use and/or computation time.
- Other numerical optimisation algorithms may be more efficient or have better convergence properties. For instance, the Gauss-Newton and Levenberg-Marquardt methods are specifically designed for optimising the squared norm of a vector-valued function, but may require a large amount of memory [3, Section 10.3].

References

[1] Pieter De Meutter and Andy W Delcloo. "Uncertainty quantification of atmospheric transport and dispersion modelling using ensembles for CTBT verification applications". In: *Journal of Environmental Radioactivity* 250 (2022), p. 106918.

[2] Gene H Golub and Charles F Van Loan. *Matrix computations*. Vol. 3. Baltimore: JHU press, 2013.

[3] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. 2nd ed. Series Title: Springer Series in Operations Research and Financial Engineering Publication Title: Numerical Optimization. New York: Springer New York, 2006. ISBN: 978-0-387-30303-1.

[4] Pauli Virtanen et al. "SciPy 1.0: fundamental algorithms for scientific computing in Python". In: *Nature methods* 17.3 (2020), pp. 261–272.