

Notes

SCK team

1 Notation

- Average emission per source: x_1, \dots, x_m where $m = 200$.
- Measured concentrations y_1, \dots, y_n where $n = 4636$. These are consecutive measurements for several measurement stations.
- $M_{i,(j,\Delta)}$ is a block in the sensitivity matrix M that represents the contribution of x_j from Δ days ago to the observed concentration y_i .
- Scaling factors s_1, \dots, s_m (in `scalings.csv`). These are put in a diagonal matrix S .
- $\mathbf{1}_q = [1 \dots 1]^T \in \mathbb{R}^q$ is the constant vector consisting of only ones.
- \odot is the Hadamard (i.e. elementwise) product, given by $(AB)_{ij} = A_{ij}B_{ij}$.
- $\text{vec} : \mathbb{R}^{p \times q} \rightarrow \mathbb{R}^{pq}$ 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.,

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

- \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 $\text{vec}(AXB) = (B^T \otimes A)\text{vec}X$ 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 Initial model

With $n = 1, \dots, 4636$ and $m = 1, \dots, 200$, the initial model is

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

That is, for each emission source j , we simulate the contribution of 15 days, and add this together for all j . A matrix formulation of this reads as follows:

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

Now we present a different interpretation. Instead of $S^{-1}x \otimes \mathbf{1}_{15}$, we make a vector $\tilde{x} := S^{-1}x \otimes \mathbf{1}_{365}$. This is a vector of length 365×200 that gives the emission from day 1 to day 365 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 365.

We can write a system of equations that is logically the same as eq. (1) but uses \tilde{x} . 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 a linear relationship that looks something like this:

$$\hat{y}_i = [\cdots \quad M_{i,(1,1)} \quad \cdots \quad M_{i,(1,15)} \quad 0 \quad \cdots \quad M_{i,(2,1)} \quad \cdots \quad M_{i,(2,15)} \quad \cdots] (S^{-1}x \otimes \mathbf{1}_{365}).$$

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

Now assume that in each station, there is one observation per day, named y_{kt} , where $k = 1, \dots, K$ where K is the number of stations and $t = 1, \dots, 365$. The index $t = 1$ corresponds to the most recent measurement. Then the full system reads as follows:

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

where

$$\tilde{x} := S^{-1}x \otimes \mathbf{1}_{365} = \begin{bmatrix} s_1^{-1}x_1 \\ \vdots \\ s_1^{-1}x_1 \\ s_2^{-1}x_2 \\ \vdots \\ s_m^{-1}x_m \end{bmatrix}.$$

We could reorganise the vector \tilde{x} to the transposed expression $\mathbf{1}_{365} \otimes (S^{-1}x)$, which 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. This has the effect of permuting the rows and columns of M_{shift} . A permutation of the variables gives a model that is essentially the same as eq. (2) but may have advantages when it comes to implementation or computational efficiency. We will not consider this.

2.2 Model with time-dependent correction to the emissions

If we weigh each day and each emission source by some factor, we are applying a transformation

$$\begin{bmatrix} s_1^{-1} x_1 \\ \vdots \\ s_1^{-1} x_1 \\ s_2^{-1} x_2 \\ \vdots \\ s_m^{-1} x_m \end{bmatrix} \mapsto \begin{bmatrix} w_{1,1} s_1^{-1} x_1 \\ \vdots \\ w_{1,365} s_1^{-1} x_1 \\ w_{2,1} s_2^{-1} x_2 \\ \vdots \\ w_{m,365} s_m^{-1} x_m \end{bmatrix}$$

which can be written concisely as

$$W(S^{-1}x \otimes \mathbf{1}_{365}) \quad \text{where} \quad W = \begin{bmatrix} w_{1,1} & 0 & \cdots & 0 \\ 0 & w_{1,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & w_{m,365} \end{bmatrix}.$$

Using this weighted vector of emissions instead of \tilde{x} in eq. (2) gives

$$\hat{y} = M_{shift} W(S^{-1}x \otimes \mathbf{1}_{365}). \quad (3)$$

In this expression, the unknown is W and should be fit 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 $\sum_{i=1}^n \log^2(h(\hat{y}_i)) = \|\log h(\hat{y})\|^2$. 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 norm¹.

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

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

$$f(W) := \underbrace{\frac{1}{2} \|\log h(\hat{y}(W))\|^2}_{\text{prediction error}} + \underbrace{\frac{1}{2} \lambda \|\log W\|^2}_{\text{deviation from average emission}} \quad \text{where} \quad \hat{y}(W) = M_{shift} W \tilde{x} \quad (4)$$

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. (4) predicts the data at least as accurately as the original model.

To optimise f numerically, we will parametrise W in eq. (4) as $W = e^{\text{diag } v}$ where $v = [v_1, v_2, \dots]$ is some vector and $e^{\text{diag } v}$ 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 $v_j \rightarrow -\infty$ 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 $W_{k+1} = W_k e^{\text{diag } \Delta v}$.

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

$$\min_v f(v) \quad \text{where} \quad f(v) = \frac{1}{2} \|\log h(\hat{y})\|^2 + \frac{1}{2} \lambda \|v\|^2 \quad \text{and} \quad \hat{y} = M_{shift} e^{\text{diag } v} \tilde{x}. \quad (5)$$

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

Proposition 1. *Let $g = [g_1, \dots, g_n]$ be the vector defined elementwise by*

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

Then

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

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

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

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

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

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

so that

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

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

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

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

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

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 $M_{\text{shift}}^T g \tilde{x}^T$ is $(M_{\text{shift}}^T g) \odot \tilde{x}$, the above simplifies to

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

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

With this information, we can optimise eq. (5) using any gradient-based optimisation method. We decided on limited-memory BFGS [3], which is implemented in SciPy [4].

5 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. (4). 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 $j = 1, \dots, m$. 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 $t+1, t+2, \dots$. We can add a constraint on the autocorrelation.
- Time-correlation could be measured by parameterising the model as follows:

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

with a constraint like $\alpha_{\min} \leq \prod_{t=1}^T \alpha_t^{(j)} \leq \alpha_{\max}$ for all T and all $j = 1, \dots, m$. 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.