STATISTICAL-PHYSICAL ESTIMATION OF POLLUTION EMISSION

Youngdeok Hwang, Emre Barut and Kyongmin Yeo

IBM Thomas J. Watson Research Center and George Washington University

Supplementary Material

S1 Generation of wind field data

The wind velocity in the s_1-direction is

$$u_1(\boldsymbol{s},t) = Real\left(\sum_{l=1}^{4}\sum_{m=1}^{4}(\widehat{u}_r(l,m,t) + i\widehat{u}_i(l,m,t))e^{i\boldsymbol{k}(l,m)\cdot\boldsymbol{s}}\right), \qquad \text{(S1.1)}$$

in which $\boldsymbol{k}(l,m) = (\frac{2\pi}{40\ \text{km}}l, \frac{2\pi}{40\ \text{km}}m)$ is the wavenumber. The Fourier coefficients are computed by solving the following stochastic differential equation;

$$\delta\widehat{u}_{r,i}(l,m,t) = -\frac{\widehat{u}_{r,i}(l,m,t)}{T_L}\delta t + S(l,m)\delta W, \qquad \text{(S1.2)}$$

in which T_L is a relaxation timescale, S is a scale parameter, and W denotes the Wiener process. Note that the solution of the stochastic differential equation is given by the Ornstein-Uhlenbeck process. In this study, $T_L = 6$

hours and $S = 2\sqrt{\frac{2}{T_L(l^2+m^2)}}$ are used. The velocity in the s_2-direction is computed from the mass conservation constraint;

$$\frac{\partial u_1(\boldsymbol{s},t)}{\partial s_1} + \frac{\partial u_2(\boldsymbol{s},t)}{\partial s_2} = 0. \tag{S1.3}$$

In other words,

$$u_2(\boldsymbol{s},t) = -Real\left(\sum_{l=1}^{p}\sum_{m=1}^{p}\frac{L_2}{L_1}\frac{l}{m}(\widehat{u}_r(l,m,t) + i\widehat{u}_i(l,m,t))e^{i\boldsymbol{k}(l,m)\cdot\boldsymbol{s}}\right). \tag{S1.4}$$

The diffusivity matrix $\boldsymbol{K}(\boldsymbol{s},t;\boldsymbol{u})$ is computed by an isotropic Smagorinsky model, which is typically used in the atmospheric dynamics models (Byun and Schere, 2006);

$$K_{ij}(\boldsymbol{s},t) = K_h(\boldsymbol{s},t)\delta_{ij},$$
$$K_h(\boldsymbol{s},t) = (C_s\Delta)^2\sqrt{\left(\frac{\partial u_1(\boldsymbol{s},t)}{\partial s_1} - \frac{\partial u_2(\boldsymbol{s},t)}{\partial s_2}\right) + \left(\frac{\partial u_1(\boldsymbol{s},t)}{\partial s_2} + \frac{\partial u_2(\boldsymbol{s},t)}{\partial s_1}\right)}.$$

Here, δ_{ij} is the Kronecker delta, $C_s(= 0.1)$ is the Smagorinsky coefficient, and the length scale $\Delta = 40km/2\pi \times 4$.

S2 Derivation of ADMM Algorithm

We define new variables, $\boldsymbol{\beta}_{\text{mse}}, \boldsymbol{\beta}_{\text{gl}}, \boldsymbol{\beta}_{\text{nuc}}, \boldsymbol{\beta}_{\text{nn}}$ and rewrite the main objective function:

$$\text{minimize} \quad f_{\text{mse}}(\boldsymbol{\beta}_{\text{mse}}) + f_{\text{gl}}(\boldsymbol{\beta}_{\text{gl}}) + f_{\text{nuc}}(\boldsymbol{\beta}_{\text{nuc}}) + f_{nn}(\boldsymbol{\beta}_{\text{nn}}) \tag{S2.5}$$

$$\text{subject to} \quad \boldsymbol{\beta}_{\text{mse}} = \boldsymbol{\beta}_{\text{gl}} = \boldsymbol{\beta}_{\text{nuc}} = \boldsymbol{\beta}_{\text{nn}} \tag{S2.6}$$

We start from a feasible solution, $\boldsymbol{\beta}_{\text{mse}} = \boldsymbol{\beta}_{\text{gl}} = \boldsymbol{\beta}_{\text{nuc}} = \boldsymbol{\beta}_{\text{nn}} = 0$. At each stage of the algorithm, we separately minimize the components of (S2.5) using proximal methods. Let $\boldsymbol{\beta}_{\text{mse}}^{(m)}$ denote the solution for minimizing f_{mse} at m^{th} iteration; we similarly define $\boldsymbol{\beta}_{g}^{(m)}$ for $g = \{\text{gl}, \text{nuc}, \text{pos}\}$. Once m^{th} iteration is complete, the average of each update can be defined to be

$$\bar{\boldsymbol{\beta}}^{m} = \left(\boldsymbol{\beta}_{\text{mse}}^{(m)} + \boldsymbol{\beta}_{\text{gl}}^{(m)} + \boldsymbol{\beta}_{\text{nuc}}^{(m)} + \boldsymbol{\beta}_{\text{nn}}^{(m)}\right)/4.$$

The dual variables, $\boldsymbol{u}_{g}^{(m)}$ for $g = \{\text{mse}, \text{gl}, \text{nuc}, \text{nn}\}$ give the deviation of $\boldsymbol{\beta}_{g}^{(m)}$ from the average. The algorithm merges $\boldsymbol{\beta}_{g}^{(m)}$ toward $\bar{\boldsymbol{\beta}}^{(m)}$ by making $\boldsymbol{u}_{g}^{(m)}$ close to zero.

At the m^{th} step the algorithm first minimizes the separate functions. For $g = \{\text{mse, gl, nuc}\}$, we set

$$\boldsymbol{\beta}_{g}^{(m+1)} \leftarrow \arg\min_{\boldsymbol{\beta}} \left(f_{g}(\boldsymbol{\beta}) + (\boldsymbol{u}_{g}^{(m)})^{\top}\left(\boldsymbol{\beta} - \bar{\boldsymbol{\beta}}^{(m)}\right) + \tfrac{1}{2\rho}\|\boldsymbol{\beta} - \bar{\boldsymbol{\beta}}^{(m)}\|_{F}^{2}\right), \tag{S2.7}$$

where ρ is a step size for the algorithm. The minimization over f_{nn} is done via projection:

$$\boldsymbol{\beta}_{\text{nn}}^{(m+1)} \leftarrow \max\{0, \bar{\boldsymbol{\beta}}^{(m)} - \boldsymbol{u}_{\text{nn}}^{(m)}\}, \tag{S2.8}$$

where max refers to the component-wise maximum.

Finally, we compute the average consensus variable and update the dual variables:

$$\begin{aligned} \bar{\boldsymbol{\beta}}^{(m+1)} &\leftarrow \left(\boldsymbol{\beta}_{\text{mse}}^{(m+1)} + \boldsymbol{\beta}_{\text{gl}}^{(m+1)} + \boldsymbol{\beta}_{\text{nuc}}^{(m+1)} + \boldsymbol{\beta}_{\text{nn}}^{(m+1)}\right)/4, \\ \boldsymbol{u}_g^{(m+1)} &\leftarrow \boldsymbol{u}_g^{(m)} + \left(\boldsymbol{\beta}_g^{(m+1)} - \bar{\boldsymbol{\beta}}^{(m+1)}\right) \qquad \text{for } g = \{\text{mse}, \text{gl}, \text{nuc}, \text{nn}\}. \end{aligned} \tag{S2.9}$$

Now, we analyze the solutions of equation (S2.7) for each component in detail.

The first function, f_{mse}, is for updating $\boldsymbol{\beta}$ with respect to the sum of prediction errors. For $k = 1, \ldots, 24$, and $i = 1, \ldots, n$, we define the matrices $\mathbf{X}(i,k) \in \mathbb{R}^{(T/24)\times p}$ as the emissions to location i at hour k, that is

$$\mathbf{X}_{t,j}(i,k) = \{X_{t,ij} : (t \mod 24) = k - 1\}.$$

We then combine these matrices row-wise and define $\mathbf{X}(k)$ for $k = 1, \ldots, 24$ as

$$\mathbf{X}(k)^\top = [\mathbf{X}(1,k)^\top, \ldots, \mathbf{X}(n,k)^\top].$$

Similarly, we define $\mathbf{y}(i,k) \in \mathbb{R}^{T/24\times 1}$ as the pollution levels at sensor i for hour k. We then set, $\mathbf{y}(k)^\top = [\mathbf{y}(1,k)^\top, \ldots, \mathbf{y}(n,k)^\top]$, i.e. all

observations from all sensors for hour k. Then, according to (S2.7),

$$\boldsymbol{\beta}_{\text{mse}}^{(m+1)} = \arg\min_{\boldsymbol{\beta}} \left(\sum_{k=1}^{24} \|\mathbf{y}(k) - \mathbf{X}(k)\boldsymbol{\beta}_{:,k}\|_2^2 + (\boldsymbol{u}_{\text{mse}}^m)^\top (\boldsymbol{\beta} - \bar{\boldsymbol{\beta}}^{(m)}) + \frac{1}{2\rho}\|\boldsymbol{\beta} - \bar{\boldsymbol{\beta}}^{(m)}\|_F^2 \right),$$

where $\boldsymbol{\beta}_{:,k}$ is the k^{th} column of $\boldsymbol{\beta}$. Note that the problem is separable over k, and differentiating with respect to $\boldsymbol{\beta}_{:,k}$ reduces to solving normal equations given by

$$\left(\mathbf{X}(k)^\top \mathbf{X}(k) + \frac{1}{2\rho}\mathbf{I}_{p\times p} \right) \boldsymbol{\beta}_{:,k} = \mathbf{X}(k)^\top \mathbf{y}(k) + \frac{1}{2\rho}\left(\bar{\boldsymbol{\beta}}_{:,k}^{(m)} - \rho \boldsymbol{u}_{\text{mse}}^{(m)}(k) \right),$$

where with an abuse of notation $\boldsymbol{u}_{\text{mse}}^{(m)}(k)$ refers to k^{th} column of $\boldsymbol{u}_{\text{mse}}^{(m)}$.

The proximal step for $\boldsymbol{\beta}_{\text{gl}}$ is given by a soft-thresholding shrinkage operator for the group lasso. That is, $\boldsymbol{\beta}_{\text{gl}}^{(m+1)}$ is obtained by

$$\boldsymbol{\beta}_{\text{gl}}^{(m+1)} \leftarrow \arg\min_{\boldsymbol{\beta}} \left(\lambda_{\text{gl}}\|\boldsymbol{\beta}\|_2 + \boldsymbol{u}_{\text{gl}}^{(m)}(\boldsymbol{\beta} - \bar{\boldsymbol{\beta}}^{(m)}) + \frac{1}{2\rho}\|\boldsymbol{\beta} - \bar{\boldsymbol{\beta}}^{(m)}\|_F^2 \right).$$

Note that, the Karush-Kuhn-Tucker (KKT) conditions for optimality are given by,

$$0 \subseteq \frac{\lambda_{\text{gl}}}{\|\boldsymbol{\beta}\|_2} + \boldsymbol{u}_{\text{gl}}^{(m)} + \frac{1}{\rho}(\boldsymbol{\beta} - \bar{\boldsymbol{\beta}}^{(m)}),$$

which gives

$$\boldsymbol{\beta}_{\text{gl}}^{(m+1)} = \begin{cases} \text{Sign}(\bar{\boldsymbol{\beta}}^{(m)} - \rho\boldsymbol{u}_{\text{gl}}^{(m)}) \left(1 - \frac{\lambda_{\text{gl}}\rho}{\left\| \bar{\boldsymbol{\beta}}^{(m)} - \rho\boldsymbol{u}_{\text{gl}}^{(m)} \right\|_2} \right) \left(\bar{\boldsymbol{\beta}}^{(m)} - \rho\boldsymbol{u}_{\text{gl}}^{(m)} \right) & \text{if } \|\bar{\boldsymbol{\beta}}^{(m)} - \boldsymbol{u}_{\text{gl}}^{(m)}\|_2 \geq \lambda_{\text{gl}}\rho, \\ 0 & \text{otherwise.} \end{cases}$$

In other words, this step applies a hard-thresholding operator on the groups, which are given by pollution sites.

For the nuclear norm regularization, f_{nuc}, consider the SVD of $\boldsymbol{\beta}^{(m)} - \boldsymbol{u}_{\text{nuc}}^{(m)} = \boldsymbol{U}^{(m)}\boldsymbol{\Sigma}^{(m)}{\boldsymbol{V}^{(m)}}^{\top}$. Then, $\boldsymbol{\beta}_{\text{nuc}}^{(m+1)}$ is given by (Theorem 2.1, Cai et al., 2010)

$$\boldsymbol{\beta}_{\text{nuc}}^{(m+1)} = \boldsymbol{U}^{(m)}\tilde{\boldsymbol{\Sigma}}^{(m)}{\boldsymbol{V}^{(m)}}^{\top},$$

where $\tilde{\boldsymbol{\Sigma}}^{(m)} = (\boldsymbol{\Sigma}^{(m)} - \lambda_{\text{nuc}}\rho\mathbb{I}_{p\times p})_+$, and $(\cdot)_+$ is applied element-wise with $(z)_+ = \max(0, z)$.

S3 Configurations for Nuclear Norm "Toy Example" Simulation

For the toy example simulation in Section 3, we generate observations from the multivariate linear regression equation,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon,$$

where $\mathbf{y} \in \mathbb{R}^{T\times 24}, \mathbf{X} \in \mathbb{R}^{T\times p}$ and $\boldsymbol{\beta} \in \mathbb{R}^{p\times 24}$ are the response, predictor and coefficient matrices, respectively.

The true coefficient matrix, $\boldsymbol{\beta}$, contains three linearly independent rows, which are given by $\beta_{\text{Type1}}, \beta_{\text{Type2}}, \beta_{\text{Type3}} \in \mathbb{R}^{1\times 24}$. Rest of the rows of $\boldsymbol{\beta}$ are given by copies of these vectors. We repeat each vector four times; hence $\boldsymbol{\beta}$ contains 12 rows and has matrix rank 3.

The contents of β_{Type1} and other types for $j = 1, \ldots, 24$ are given by:

$$\begin{aligned}
\beta_{\text{Type1,j}} =& 1, \\
\beta_{\text{Type2},j} =& \frac{1.5}{144}(24(j-1) - (j-1)^2), \\
\beta_{\text{Type3},j} =& \begin{cases} 2 & \text{if } j \in \{7, 8, 9, 16, 17, 18\}, \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

We fill the entries of $\mathbf{X}$ by drawing independent copies from a Uniform[0,1] distribution. The error matrix, ϵ is similarly filled by independent copies of a standard normal distribution.

Bibliography

Byun, D. and Schere, K. L. (2006), "Review of the Governing Equations, Computational Algorithms, and Other Components of the Models-3 Community Multiscale Air Quality (CMAQ) Modeling System," *Applied Mechanics Reviews*, 59, 51–77.

Cai, J., Candès, E., and Shen, Z. (2010), "A Singular Value Thresholding Algorithm for Matrix Completion," *SIAM Journal on Optimization*, 20, 1956–1982.