Supplemental Information for

Source Apportionment of Time and Size Resolved Ambient Particulate Matter

Philip K. Hopke,[1*] Na Li,[2] Steven S, Smith,[3] Carmeliza Navasca[4]

1. Center for Air Resource Engineering and Science, Clarkson University, Box 5708, Potsdam, NY 13699 USA
2. Department of Mathematics, Clarkson University, Box 5815, Potsdam, NY 13699 USA
3. Department of Applied Science, University of California, Davis, CA 95616 USA
4. Department of Mathematics, University of Alabama at Birmingham, 1300 University Boulevard, Birmingham, AL 35294 USA

Weighted Alternating Least Squares Algorithm

Here, we introduce an iterative way to solve the receptor model (6). We need to give several definitions first.

Definition 1 (Mode-n fibers): *A mode-n fiber of a Nth order tensor is a vector defined by fixing every index but the n-th one.*

For example, a matrix column is a mode-1 fiber and a matrix row is a mode-2 fiber. Third-order tensor have column (mode-1), row (mode-2) and tube (mode-3) fibers, denoted by $\mathbf{x}_{:jk}$, $\mathbf{x}_{i:k}$, $\mathbf{x}_{ij:}$ respectively.

Definition 2 (Mode-n matricization): *Matricization is the process of reordering the elements of an Nth order tensor into a matrix. The mode-n matricization of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted by $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \cdots I_{n-1} I_{n+1} \cdots I_N)}$ and arranges the mode-n fibers to be the columns of the resulting matrix.*

Definition 3 (Vectorization): *The vectorization of a matrix $\mathbf{M} = [\mathbf{m_1} \quad \cdots \quad \mathbf{m_n}] \in \mathbb{R}^{m \times n}$, where $\mathbf{m_i}$ is the i-th column of $\mathbf{M}$, is denoted by $vec(\mathbf{M})$ which is a vector of size (mn) defined by*

$$vec(\mathbf{M}) = \begin{bmatrix} \mathbf{m_1} \\ \vdots \\ \mathbf{m_n} \end{bmatrix}.$$

* Author to whom correspondences should be directed. E-mail: phopke@clarkson.edu

Notice that the only difference between (1) and (6) is the additional uncertainty tensor, so we can work on (1) first and easily have the two variations of (6) by adding on the uncertainty.

By taking mode-3 matricization on the both sides of the original model (1), we can obtain

$$\mathbf{X}_{(3)}^T = M(\mathbf{A}) \cdot \mathbf{B}^T + \mathbf{E}_{(3)}^T, \quad (7)$$

$$M(\mathbf{A}) = [vec((\mathbf{A}^{(1)})^T) \quad \cdots \quad vec((\mathbf{A}^{(R)})^T)] \in \mathbb{R}^{IJ \times R}, \quad (8)$$

$$\mathbf{B} = [\mathbf{b}^{(1)} \quad \cdots \quad \mathbf{b}^{(R)}] \in \mathbb{R}^{K \times R}. \quad (9)$$

Similarly, we can consider mode-1 matricization on the equation (1). So it becomes

$$\mathbf{X}_{(1)}^T = (\mathbf{D} \odot_{\boldsymbol{l}} \mathbf{B}) \cdot \mathbf{A}^T + \mathbf{E}_{(1)}^T, \quad (10)$$

where $\mathbf{D}$ is a $J \times J$ identity matrix, and $\odot_{\boldsymbol{l}}$ is a Khatri-Rao like product which is defined as follows,

$$\mathbf{D} \odot_{\boldsymbol{l}} \mathbf{B} = [\mathbf{D} \otimes \mathbf{b}_1 \quad \cdots \quad \mathbf{D} \otimes \mathbf{b}_Q] \in \mathbb{R}^{JK \times JR}. \quad (11)$$

Therefore, if we matricize the uncertainty tensor $\mathcal{U}$ to get $\mathbf{U}_{(1)}$ and $\mathbf{U}_{(3)}$ we can have two variations of the cost function (6) according to the equation (7) and (10), :

$$Q1 = \left\|(\mathbf{X}_{(3)}^T - M(\mathbf{A}) \cdot \mathbf{B}^T)/\mathbf{U}_{(3)}^T\right\|_F^2, \quad (12)$$

$$Q2 = \left\|(\mathbf{X}_{(1)}^T - (\mathbf{D} \odot_{\boldsymbol{l}} \mathbf{B}) \cdot \mathbf{A}^T)/\mathbf{U}_{(1)}^T\right\|_F^2, \quad (13)$$

where the '/' denotes element-wise division between two matrices.

Given initials $\mathbf{A}^0$ and $\mathbf{B}^0$, we can update $\mathbf{A}$ by fixing the factor matrix $\mathbf{B}$ in (13), and update $\mathbf{B}$ by fixing the factor matrix $\mathbf{A}$ in (12). So the problem (6) can be solved by the following two subproblems:

$$\mathbf{A}^{k+1} = \text{argmin}_{\widehat{\mathbf{A}} \in \mathbb{R}^{I \times JR}} \left\|(\mathbf{X}_{(1)}^T - (\mathbf{D} \odot_{\boldsymbol{l}} \mathbf{B}^k) \cdot \widehat{\mathbf{A}}^T)/\mathbf{U}_{(1)}^T\right\|_F^2 \quad (14)$$

$$\mathbf{B}^{k+1} = \text{argmin}_{\widehat{\mathbf{B}} \in \mathbb{R}^{K \times R}} \left\|(\mathbf{X}_{(3)}^T - M(\mathbf{A}^{k+1}) \cdot \widehat{\mathbf{B}}^T)/\mathbf{U}_{(3)}^T\right\|_F^2 \quad (15)$$

where $\mathbf{A}^{k+1}$ and $\mathbf{B}^{k+1}$ are the results obtained at the $(k+1)$th iteration.

In [14], the problem is to minimize the following cost function

$$O = \|(\mathbf{X} - \mathbf{CP})/\mathbf{\Sigma}\|_F^2$$

where $\mathbf{X}, \mathbf{\Sigma} \in \mathbb{R}^{I \times J}$, $\mathbf{C} \in \mathbb{R}^{I \times R}$ and $\mathbf{P} \in \mathbb{R}^{R \times J}$. So the WALS algorithm proposed in [14] is

$$\mathbf{C}^{k+1} = \text{argmin}_{\widehat{\mathbf{C}} \in \mathbb{R}^{I \times R}} \left\|(\mathbf{X} - \widehat{\mathbf{C}}\mathbf{P}^k)/\mathbf{\Sigma}\right\|_F^2,$$

$$\mathbf{P}^{k+1} = \text{argmin}_{\widehat{\mathbf{P}} \in \mathbb{R}^{R \times J}} \left\|(\mathbf{X} - \mathbf{C}^{k+1}\widehat{\mathbf{P}})/\mathbf{\Sigma}\right\|_F^2.$$

As we can see that the WALS method in [14] solved the two factor matrices by using the same objective function. Alternatively we are using two objective function $Q1$ and $Q2$. The reasoning behind this is that our least squares is more complicated. The factor matrix $\mathbf{A}$ cannot easily be solved by $Q1$, and similarly $\mathbf{B}$ cannot be solved by $Q2$ directly.

So our task is to solve the subproblems (12) and (13) with non-negativity constraints. One way is to treat the each subproblem as a nonlinear optimization with constraints, but nonlinear solver is expensive.

Instead of solving the factor matrix once a time, we can solve it vector by vector. The subproblem (14) is used to demonstrate this process. Let $\mathbf{x}_i$ and $\mathbf{u}_i$ denote the ith columns of $\mathbf{X}_{(1)}^T$ and $\mathbf{U}_{(1)}^T$ respectively, $\hat{\mathbf{a}}_i$denote the ith column of $\widehat{\mathbf{A}}^T$. So, we have

$$\left\|(\mathbf{X}_{(1)}^T - (\mathbf{D} \odot_{\boldsymbol{l}} \mathbf{B}^k) \cdot \widehat{\mathbf{A}}^T)/\mathbf{U}_{(1)}^T\right\|_F^2 = \sum_{i=1}^{I} \|(\mathbf{x}_i - (\mathbf{D} \odot_{\boldsymbol{l}} \mathbf{B}^k) \cdot \hat{\mathbf{a}}_i)/\mathbf{u}_i\|_2^2 \tag{16}$$

For each $\mathbf{u}_i, i = 1,2, \dots, I$, we define a matrix $\mathbf{Du}_i$ as

$$\mathbf{Du}_i = \begin{bmatrix} 1/(\mathbf{u}_i)_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1/(\mathbf{u}_i)_1 \end{bmatrix} \in \mathbb{R}^{JK \times JK}, \tag{17}$$

Where $1/(\mathbf{u}_i)_s$ is the sth element of the vector $\mathbf{u}_i$, $s = 1,2, \dots, JK$.

Therefore, for the each item in the summation, we have

$$\begin{aligned} &\|(\mathbf{x}_i - (\mathbf{D} \odot_{\boldsymbol{l}} \mathbf{B}^k) \cdot \hat{\mathbf{a}}_i)/\mathbf{u}_i\|_2^2 \\ &= (\mathbf{x}_i - (\mathbf{D} \odot_{\boldsymbol{l}} \mathbf{B}^k) \cdot \hat{\mathbf{a}}_i)^T \cdot \mathbf{Du}_i{}^2 \cdot (\mathbf{x}_i - (\mathbf{D} \odot_{\boldsymbol{l}} \mathbf{B}^k) \cdot \hat{\mathbf{a}}_i) \\ &= (\mathbf{Du}_i(\mathbf{x}_i - (\mathbf{D} \odot_{\boldsymbol{l}} \mathbf{B}^k) \cdot \hat{\mathbf{a}}_i))^T \cdot (\mathbf{Du}_i(\mathbf{x}_i - (\mathbf{D} \odot_{\boldsymbol{l}} \mathbf{B}^k) \cdot \hat{\mathbf{a}}_i)) \\ &= \|\mathbf{Du}_i\mathbf{x}_i - \mathbf{Du}_i(\mathbf{D} \odot_{\boldsymbol{l}} \mathbf{B}^k) \cdot \hat{\mathbf{a}}_i\|_2^2 \end{aligned}$$

and problem is to minimize the following function,

$$\min_{\hat{\mathbf{a}}_i} \|\mathbf{Du}_i\mathbf{x}_i - \mathbf{Du}_i(\mathbf{D} \odot_{\boldsymbol{l}} \mathbf{B}^k) \cdot \hat{\mathbf{a}}_i\|_2^2 \tag{18}$$

Noticed that the equation (18) is just a least-squares problem with the coefficient matrix $\mathbf{Du}_i(\mathbf{D} \odot_{\boldsymbol{l}} \mathbf{B}^k)$, where $\mathbf{Du}_i$ is defined by (17). So, instead of solving the whole matrix $\widehat{\mathbf{A}}^T$ once a time, we can calculate each column $\hat{\mathbf{a}}_i$ by (18). In addition, minimizing the objective function (12) in terms of $\hat{\mathbf{a}}_i$ with non-negativity constraint is easy to solve in MATLAB. We will use standard MATLAB least squares function 'lsqnonneg', to impose the non-negativity constraints.

To stop the algorithm, we need to provide a convergence criterion. Usually the sum of squared residual (SSR) is used as convergence criterion. This criterion stops the algorithm if the

change of objective Q between two iterations is less than some small number called the tolerance. However, this may not be the best convergence criterion to cope with our problem.

In [19], a new convergence criterion is introduced. It keeps tracking the change in $\det(\mathbf{B}^T\mathbf{B})$ from one iteration to the next. Where $\mathbf{B}$ is defined by (9) and 'det' is the determinant of the given matrix. $\det(\mathbf{B}^T\mathbf{B})$ is the squared volume of the space spanned by the column space of **B,** it has advantages in the resolution process.

Therefore, the algorithm will stop if both of the changes (the value of objective function Q and $\det(\mathbf{B}^T\mathbf{B})$) between two iterations are sufficiently small (~10^{-8}).

We summarize the algorithm as follows:

WALS-Algorithm

Input: Tensors $\mathcal{X},\mathcal{U} \in \mathbb{R}^{I\times J\times K}$, factor numbers R, $\mathbf{A}^0 \in \mathbb{R}^{I\times JR}$, $\mathbf{B}^0 \in \mathbb{R}^{K\times R}$
Output: Nonegative matrices $\mathbf{A} \in \mathbb{R}^{I\times JR}$, $\mathbf{B} \in \mathbb{R}^{K\times R}$ minimize (12) and (13).

$k = 0$
$\mathbf{U1} = \mathbf{U}_{(1)}^T$, $\mathbf{U3} = \mathbf{U}_{(3)}^T$, $\mathbf{X1} = \mathbf{X}_{(1)}^T$, $\mathbf{X3} = \mathbf{X}_{(3)}^T$, $\mathbf{D} = \text{eye}(J)$
$CC1 = 1$, $CC2 = 1$
while $CC1 > tol$ and $CC2 > tol$ **do**
  for $i = 1,\ldots,I$ **do**
    $\mathbf{Du} = \text{diag}(1./\mathbf{U1}(:,i))$
    $\mathbf{A}^{k+1}(i,:)^T = \text{lsqnonneg}\ (\mathbf{Du}\cdot(\mathbf{D}\odot_l \mathbf{B}^k), \mathbf{Du}\cdot\mathbf{X1}(:,i)))$
  end for
  for $j = 1,\ldots,K$ **do**
    $\mathbf{Du} = \text{diag}(1./\mathbf{U3}(:,i))$
    $\mathbf{B}^{k+1}(j,:)^T = \text{lsqnonneg}\ (\mathbf{Du}\cdot M(\mathbf{A}^{k+1}), \mathbf{Du}\cdot\mathbf{X3}(:,j))$
  end for
  $CC1 = \left\|(\mathbf{X3} - M(\mathbf{A}^{k+1})\mathbf{B}^{k+1})./\mathbf{U3}\right\|_F^2$
  $CC2 = \left|\det((\mathbf{B}^k)^T\mathbf{B}^k) - ((\mathbf{B}^{k+1})^T\mathbf{B}^{k+1})\right|$
  $k = k + 1$
end while
$\mathbf{A} = \mathbf{A}^k, \mathbf{B} = \mathbf{B}^k$

In Section 3, we introduced the BTD-$(L; L; 1)$ model and pointed out it is an another way around of the receptor model. So we could theoretically use the BTD-$(L; L; 1)$ algorithm to solve the problem. On the other hand, it is important to note several advantages in using WALS. First, additional decomposition in $\mathbf{A}_r \cdot \mathbf{B}_r^T$ for each r by using BTD-$(L; L; 1)$ does not give additional information due to lack of physical interpretation. Second, WALS method is much faster than BTD-$(L; L; 1)$ because it only solve two factors in each iteration. Lastly, we can easily add

additional constraints on the WALS algorithm based on some prior information. For example, we need constrain both matrices $\mathbf{A}_r$ and $\mathbf{B}_r$ in BTD-$(L; L; 1)$ in order to add some constraint on the profile matrix.