https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

Positive Matrix Factorization of Large Aerosol Mass Spectrometry Datasets Using Error-Weighted Randomized Hierarchical Alternating Least Squares

Benjamin Sapper[1], Daven Henze[1], Manjula Canagaratna[2], and Harald Stark[3,4]

[1]University of Colorado Boulder, 11 Engineering Dr, Boulder, CO 80309, United States
[2]Aerodyne Research, 45 Manning Road, Billerica, MA 01821, United States
[3]Center for Aerosol and Cloud Chemistry, Aerodyne Research Inc., 45 Manning Road, Billerica, MA 01821
[4]Department of Chemistry and Cooperative Institute for Research in Environmental Sciences (CIRES), University of Colorado, Boulder, Colorado 80309, United States

Correspondence: Benjamin Sapper (bsapper77@gmail.com)

Abstract. Weighted positive matrix factorization (PMF) has been used by scientists to find small sets of underlying factors in environmental data. However, as the size of the data has grown, increasing computational costs have made it impractical to use traditional methods for this factorization. In this paper, we present a new weighting method to dramatically decrease computational costs for these traditional algorithms. We then apply this weighting method with the Randomized Hierarchical Alternating Least Squares (RHALS) algorithm to a large environmental dataset, where we show that interpretable factors can be reproduced using these methods. We show this algorithm results in a computational speedup of 38, 67, and 634 compared to the Multiplicative Update (MU), deterministic Hierarchical Alternating Least Squares (HALS), and non-negative Alternating Least Squares (ALS) algorithms, respectively. We also investigate rotational ambiguity in the solution, and present a simple "pulling" method to rotate a set of factors. This method is shown to find alternative solutions, and in some cases, lower the weighted residual error of the algorithm.

1 Introduction

1.1 Problem Statement

Low rank matrix factorization has been widely used in data science to explain underlying factors in large datasets (Xie et al. (1998), Kim and Hopke (2007), Wei et al. (2016)). The process considers a data matrix, $\mathbf{A}$, of size $m \times n$, that is decomposed into two smaller matrices, $\mathbf{W}$ of size $m \times k$ and $\mathbf{H}$ of size $k \times n$, where $k \ll m, n$ and $\mathbf{A} \approx \mathbf{WH}$. Traditionally, principal component analysis (PCA) and singular value decomposition (SVD) have been used to find these factors (Kumar (2017), Wei et al. (2016)). PCA finds the eigenvectors of the covariance matrix $\mathbf{A}^T\mathbf{A}$, which are called the principal components, representing the directions of maximum variance in the data. The vectors are ordered by how much variance they explain, and only the most important vectors are kept, which are identified as the underlying factors in the data. Closely related, the SVD finds the factorization $\mathbf{A} = \mathbf{USV}^T$, where $\mathbf{U}$ and $\mathbf{V}$ contain the left and right singular vectors, respectively, and $\mathbf{S}$ is a diagonal

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

matrix containing the singular values of $\mathbf{A}$ in decreasing order. Thus, to find a low rank approximation of $\mathbf{A}$, one could keep only the k most significant singular values and vectors to form the truncated SVD, $\mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^T$. Mathematically, the truncated SVD is the most optimal rank k factorization of $\mathbf{A}$ for minimizing squared error (Eckart and Young (1936)). However, the SVD is not optimal for some factorization problems in several ways (Paatero and Tapper (1994)):

1. The SVD produces factors with negative values: for some factor analysis problems, such as finding chemical sources to air pollution data, SVD results are hard to interpret because chemical concentrations can only be positive.

2. The SVD produces orthogonal factors: most factor analysis problems do not carry the underlying constraint that the factors have to be orthogonal to each other.

3. The SVD is not fit to solve the following weighted least squares problem: suppose that accompanying the dataset $\mathbf{A}$ is an equally sized matrix $\mathbf{\Sigma}$ with $\mathbf{\Sigma}_{ij} = \sigma_{ij}$ representing the uncertainty of the measurement for $\mathbf{A}_{ij}$. If $rank(\mathbf{\Sigma}) > 1$, the SVD can't be scaled to find a solution minimizing the weighted residual error, which is defined as $\sum_i \sum_j \frac{(\mathbf{A}_{ij} - \sum_k \mathbf{W}_{ik}\mathbf{H}_{kj})^2}{\sigma_{ij}^2}$ (Paatero and Tapper (1994)).

Thus, positive matrix factorization (PMF) was introduced by Paatero to address these concerns (Paatero and Tapper (1993)). Weighted PMF attempts to find two factor matrices $\mathbf{W}$ and $\mathbf{H}$ by minimizing the equation

$$\min_{\mathbf{W},\mathbf{H}} || \frac{\mathbf{A} - \mathbf{WH}}{\mathbf{\Sigma}} ||_F^2 \text{ with } \mathbf{W} \geq 0, \mathbf{H} \geq 0 \quad (1)$$

In Eq. (1), the division by $\mathbf{\Sigma}$ represents elementwise division, the norm $||||_F$ is the Frobenius norm, and all elements of $\mathbf{W}$ and $\mathbf{H}$ are constrained to be positive.

Traditional factor analysis methods are known to be computationally expensive. Steps to speed up factor analysis have thus been explored, such as randomization and the use of GPUs (Halko et al. (2011), Tan and Cao). Recently, the authors in (Erichson et al. (2018)) applied randomization to PMF, introducing a new method, Randomized Hierarchical Alternating Least Squares (RHALS) to solving the unweighted PMF problem. In this paper, we test the application of RHALS to atmospheric concentration data that contain uncertainties. Accounting for these uncertainties as regression weights, we introduce a method of externally weighting and unweighting the data, which to our knowledge is novel in its application to RHALS. We consider the accuracy and the reduced computational costs compared to other PMF algorithms commonly used in the field of atmospheric science.

1.2 Background

1.2.1 PMF2 and Paatero

The first widely accepted algorithm for PMF was derived by Paatero using the Gauss-Newton Method (Paatero (1997)). This algorithm, called PMF2, is very commonly used with environmental data (Kim and Hopke (2007), Ulbrich et al. (2009), Massoli et al. (2018)). Paatero defines an enhanced objective function, and attempts to find factor matrices $\mathbf{W}$ and $\mathbf{H}$ that minimize Q in

$$Q = \sum_i \sum_j \frac{(\mathbf{A}_{ij} - \sum_k \mathbf{W}_{ik}\mathbf{H}_{kj})^2}{\sigma_{ij}^2} - \alpha \sum_i \sum_k \log(\mathbf{W}_{ik}) - \beta \sum_k \sum_j \log(\mathbf{H}_{kj}) + \quad \gamma \sum_i \sum_k \mathbf{W}_{ik}^2 + \delta \sum_k \sum_j \mathbf{H}_{kj}^2 \quad (2)$$

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

In Eq. (2), logarithmic penalty terms are added to penalize factor values that become too close to zero (and therefore potentially negative), and L2 regularization is added to smooth out the factors and avoid overfitting (Paatero (1997)). α and β control the strength of the penalty terms, while γ amd δ control the strength of L2 regularization. While Paatero hasn't released the source code detailing the exact algorithm to solve Eq. (2), a step-by-step guide in pseudocode as to how to use the Gauss-Newton method to solve Eq. (2) is detailed in (Lu and Wu (2004)).

1.2.2 Multiplicative Update

An alternative method for PMF was developed in (Lee and Seung (1999)). The multiplicative update (MU) method utilizes a special case of gradient descent where the learning rates are chosen to avoid subtraction in the gradient (Burred). A multiplicative update estimate of a parameter θ (either $\mathbf{W}$ or $\mathbf{H}$) is found by updates of the form (Burred)

$$\theta = \theta \odot \frac{\nabla_\theta^- Q(\theta)}{\nabla_\theta^+ Q(\theta)} \tag{3}$$

where $Q(\theta)$ is a cost function to be minimized, ∇_θ^- consists of the negative terms of the gradient of the cost function, ∇_θ^+ consists of the positive terms of the gradient, and $\odot$ is an element wise multiplication. Since the data matrix $\mathbf{A}$, the uncertainties, and the factor matrices are all positive at each step of this algorithm, the factor matrices in the subsequent step will also be positive because Eq. (3) only deals with the multiplication and division of positive numbers.

1.2.3 Alternating Least Squares and Hierarchical ALS

Alternating Least Squares (ALS) methods solve for the factor matrices $\mathbf{W}$ and $\mathbf{H}$ by iteratively updating each matrix until convergence is reached (Tan and Cao, Takács and Tikk (2012)). The cost function Q, containing $||\frac{\mathbf{A}-\mathbf{WH}}{\mathbf{\Sigma}}||_F^2$, is minimized by setting the partial derivatives $\frac{\partial Q}{\partial \mathbf{W}}$ and $\frac{\partial Q}{\partial \mathbf{H}}$ to zero, and solving for $\mathbf{W}$ and $\mathbf{H}$. To satisfy the positivity constraint, negative elements in the factors are set to zero.

In recent years, Hierarchical Alternating Least Squares (HALS) has become increasingly popular as an efficient method for PMF (Cichocki and Phan (2009)). Instead of minimizing with respect to the entire factor matrices $\mathbf{W}$ amd $\mathbf{H}$, HALS minimizes the cost function with respect to one block, or an outer product, of individual factors at a time. The main component of the cost function is redefined as $Q^j = ||\frac{\mathbf{R}^j - \mathbf{W}(:,j)\mathbf{H}(j,:)}{\mathbf{\Sigma}}||_F^2$, where $\mathbf{W}(:,j)$ and $\mathbf{H}(j,:)$ are the jth factors. Q^j can be minimized for each factor j by setting the partial derivatives $\frac{\partial Q^j}{\partial \mathbf{W}(:,j)}$ and $\frac{\partial Q^j}{\partial \mathbf{H}(j,:)}$ to zero and solving for $\mathbf{W}(:,j)$, and $\mathbf{H}(j,:)$.

The derivation of the ALS update rules is detailed in section A of the Appendix, while the derivation of HALS is detailed in the Methods section.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

1.3 Reducing Computational Costs

1.3.1 Randomization

To reduce the computational costs of a matrix factorization algorithm for large datasets, randomization methods have been used as a dimension reduction technique (Erichson et al. (2018), Halko et al. (2011), Kaloorazi and Chen (2019)). Below, we present a brief overview of the theory and results layed out in (Halko et al. (2011)).

When performing randomization techniques, we hope that much of the relevant information about the column space of the data matrix $\mathbf{A}$ can be stored in just a few vectors that we can sample. This is only true if the effective rank of $\mathbf{A}$ is low ($\mathbf{A}$ only has a few nonnegligible singular values), but that is generally assumed to be the case in any PMF problem (Erichson et al. (2018)). Mathematically, we seek the approximation

$$\mathbf{A} \approx \mathbf{Q}\mathbf{Q}^T\mathbf{A} \tag{4}$$

where the relatively few number of columns in $\mathbf{Q}$ are orthonormal and form an approximate basis of $\mathbf{A}$. Choosing the columns of $\mathbf{Q}$ to be the left singular vectors of A would minimize L2 error: if the first k singular vectors were chosen, the error term $||\mathbf{A} - \mathbf{Q}\mathbf{Q}^T\mathbf{A}||_2 = \sigma_{k+1}$, with σ_{k+1} being the $(k+1)^{st}$ largest singular value of $\mathbf{A}$ (Halko et al. (2011)). However, it turns out that a suitable basis can also be found from random samples of the column space of $\mathbf{A}$.

How many samples should one take from the column space of $\mathbf{A}$? Since we assume there are k underlying factors within $\mathbf{A}$, we essentially say that the effective rank of $\mathbf{A}$ is k and sample k random samples from the column space of $\mathbf{A}$. However, with underlying uncertainties, we can write $\mathbf{A} = \mathbf{B} + \mathbf{E}$, where $\mathbf{B}$ is the rank k matrix spanned by the factors for which we wish to find a basis, and $\mathbf{E}$ is a perturbation matrix filled with the noise in $\mathbf{A}$ (Halko et al. (2011)). Suppose we were to sample from the column space of A – that is, form the vector $\boldsymbol{y} = \mathbf{A}\boldsymbol{\omega} = \mathbf{B}\boldsymbol{\omega} + \mathbf{E}\boldsymbol{\omega}$. Each vector $\boldsymbol{y}$ is slightly pushed out of the column space by the term $\mathbf{E}\boldsymbol{\omega}$. Thus, to increase the likelihood of spanning the column space of $\mathbf{B}$, p more vectors are sampled from $\mathbf{A}$. In practice, choosing p to be 10 or 20 is sufficient.

To construct this low rank approximation, $k+p$ random normal samples, stored as columns of the matrix $\boldsymbol{\Omega}$ (dimensions $n \times (k+p)$) of the column space of $\mathbf{A}$ are taken and stored in $\mathbf{Y}$:

$$\underset{m\times(k+p)}{\mathbf{Y}} = \underset{m\times n}{\mathbf{A}} * \underset{n\times(k+p)}{\boldsymbol{\Omega}} \tag{5}$$

Next, the columns of $\mathbf{Y}$ are orthonormalized using a QR decomposition to form our projection matrix $\mathbf{Q}$. The algorithm can now be run on the lower dimensional matrix $\mathbf{B} = \mathbf{Q}^T\mathbf{A}$.

1.3.2 Parallelization

Many new algorithms use multiple cores within a computer, including Graphics Processing Units (GPUs), for computational efficiency (Tan and Cao, Nasre et al. (2013)). Tasks that are parallelizable, such as matrix multiplication, can be partitioned and run simultaneously on these cores, decreasing computing time by a factor of the number of cores used. While this paper

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

does not explore the advantages of parallelization on the RHALS algorithm, we discuss the parts of the algorithm that can be parallelized, in particular the randomization step and the external weighting step derived in Section 2.3.

1.4 Non-Uniqueness of Solutions

Unlike the SVD, there is no uniqueness constraint on the factor matrices **W** and **H** for PMF. That is, the factorization $\mathbf{A} = \mathbf{WH}$ can also be expressed as $\mathbf{A} = \mathbf{WTT}^{-1}\mathbf{H}$, where **T** is a rotational matrix and $\hat{\mathbf{W}} = \mathbf{WT}$ and $\hat{\mathbf{H}} = \mathbf{T}^{-1}\mathbf{H}$ are the new rotated factors. Thus, to span the space of feasible solutions, previous approaches such as PMF2 have aimed to find new solutions $\hat{\mathbf{W}}$ and $\hat{\mathbf{H}}$ by varying **T** (Paatero (1997)), or varying initializations (Ulbrich et al. (2009)).

The rotational matrix **T** is a $k \times k$ matrix where t_{ii}, a diagonal element of **T**, represents a scaling of the i^{th} factor, and t_{ij} represents a rotation of the j^{th} factor towards the i^{th} factor in **W**, and a rotation of the i^{th} factor away from the j^{th} factor in **H**. For example, consider the elementary rotation matrices

$$\mathbf{T}_E = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & r & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}, \mathbf{T}_E^{-1} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -r & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \tag{6}$$

All factors remain the same, except $\hat{\mathbf{W}}(:,3) = \mathbf{W}(:,3) + r\mathbf{W}(:,2)$, and $\hat{\mathbf{H}}(2,:) = \mathbf{H}(2,:) - r\mathbf{H}(3,:)$.

Regardless of whether r is positive or negative, values in either **W** or **H** will be pulled towards negative values. Thus if a large proportion of the factors are filled with zeros, there may be little to no pure rotations, or rotations that do not change the residual $||\mathbf{A} - \mathbf{WH}||_F^2$. Algorithms such as those developed by Paatero, where a logarithmic penalty term is added to push factor values to be more positive, will have few zeros in the factor matrices, and thus will have more rotational ambiguity (Paatero et al. (2005)). However, RHALS enforces non-negativity merely by setting negative elements to zero, and thus many values in the factor matrices may end up being zero. Thus for RHALS, there will almost certainly not exist a perfect rotational matrix **T**, and only "approximate" rotations can be studied, in which the rotation will alter the value of the weighted error.

It is not feasible to span all possible variants that **T** can take. Thus, the problem is often simplified to considering only positive rotations (values of **T** greater than zero) and negative rotations (values of **T** less than zero). A rotational program in PMF2 called FPEAK uses the parameter ϕ to denote the rotation strength, with positive values leading to positive rotations in **W** (Paatero (1997)). Paatero further improved this method in the Multilinear Engine (ME) algorithm, where the strength of rotation is allowed to vary between factors (Paatero and Hopke (2009)). The pulling algorithm presented in (Paatero and Hopke (2009)) is a sophisticated rotational method; more rudimentary pulling methods that mimic varying the regularization of the factor matrices are presented in (Paatero (1997)) and (Paatero et al. (2002)).

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

1.5 Scaling with Uncertainties

Recall that to account for inaccuracies in real data, we measure the error of the algorithm by dividing the residual by the standard deviation of the uncertainty of each measurement:

$$Q = \sum_{i=1}^{m}\sum_{j=1}^{n} \frac{\mathbf{A}_{ij} - \sum_{d=1}^{k} \mathbf{W}_{id}\mathbf{H}_{dj}}{\sigma_{ij}} \tag{7}$$

To account for these uncertainties, one could incorporate them into each update rule of the factor matrices, as is done in PMF2 (Paatero (1997)). In this paper, we refer to this as "internal weighting". However, doing this is very computationally expensive due to the size of the uncertainty matrix $\mathbf{\Sigma}$. Furthermore, how would the uncertainties be included after the dimension of the data is reduced by a randomization step? One way to address the weighting problem introduced by dimension reduction is through an expectation maximization step layed out in (Yahaya et al. (2019)). At each iteration, the data matrix $\mathbf{A}$ is scaled based on the uncertainties and current factors, and then compressed again into a lower dimension. However, this method is still computationally expensive – in fact, the computationally expensive step of expectation maximization eliminates any of the time benefits gained by randomization. We thus introduce an alternate approach to weighted PMF where the data is prescaled by the uncertainties matrix, the unweighted algorithm is applied to the scaled data, and the converged factors are scaled by the uncertainties. This approach, which we will refer to as "external weighting",thus dramatically reduces computational costs and allows for dimensionality reduction, since weights are not included in the update rules.

It is noted in (Paatero and Tapper (1993)) that if the weighting matrix is rank one, that is if $1/\sigma_{ij} = \boldsymbol{B}_i\boldsymbol{C}_j$ for vectors $\boldsymbol{B}$, $\boldsymbol{C}$, then an optimal scaling can be found. By forming the diagonal matrices $\mathbf{D}_L$, $\mathbf{D}_R$ with $[\mathbf{D}_L]_{ii} = \boldsymbol{B}_i$, $[\mathbf{D}_R]_{jj} = \boldsymbol{C}_j$, if $\mathbf{A} = \mathbf{WH} + \mathbf{E}$, then $\mathbf{D}_L\mathbf{A}\mathbf{D}_R = (\mathbf{D}_L\mathbf{W})(\mathbf{H}\mathbf{D}_R) + \mathbf{D}_L\mathbf{E}\mathbf{D}_R = \hat{\mathbf{W}}\hat{\mathbf{H}} + \hat{\mathbf{E}}$. Thus, by first finding $\hat{\mathbf{A}} = \mathbf{D}_L\mathbf{A}\mathbf{D}_R$ and then running a PMF algorithm without weights on $\hat{\mathbf{A}}$, one can produce estimated factor matrices $\hat{\mathbf{W}}$ and $\hat{\mathbf{H}}$, where the unscaled estimates are $\mathbf{W} = \mathbf{D}_L^{-1}\hat{\mathbf{W}}$ and $\mathbf{H} = \hat{\mathbf{H}}\mathbf{D}_R^{-1}$.

If the weighting matrix $1/\mathbf{\Sigma}$ isn't rank one – which is very likely for environmental data – and cannot be estimated as the outer product of two vectors, there is no scaling of the previous form that can be applied to the data matrix $\mathbf{A}$ (Paatero and Tapper (1993)). To address this, Paatero presented a simple algorithm to find an approximate rank 1 factorization of $1/\mathbf{\Sigma}$ (Paatero and Tapper (1993)). We present a different method where the data is scaled by the full rank matrix $\mathbf{\Sigma}$, and then unscaled after the algorithm is complete. This algorithm is described in detail in Section 2.3.

1.6 Determining the Number of Factors

The quality of fit parameter, defined as $\mathcal{Q} = \left(\sum_i \sum_j \frac{\mathbf{A}_{ij} - \sum_k \mathbf{W}_{ik}\mathbf{H}_{kj}}{\sigma_{ij}}\right)^2$, i.e. the squared weighted residual error, can be used to determine whether a given solution either overfits or underfits the data (Ulbrich et al. (2009)). If measurement errors are normally distributed, then $\mathcal{Q}$ will follow a χ^2 distribution with $mn - k(m+n)$ degrees of freedom. Thus, to avoid overfitting, the number of factors in a solution are chosen such that $\mathcal{Q} \approx mn - k(m+n)$ (Paatero et al. (2002)). When additional error is present, measuring the convergence of $\mathcal{Q}$ or the weighted residual error with additional factors can determine if these additional factors add much information to the model.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

Another method used in determining the number of factors is the lack of rotational ambiguity of a solution (Ulbrich et al. (2009)). Consider a simple case where the data matrix $\mathbf{A}$ is the product of two rank-two matrices $\mathbf{W} = [\boldsymbol{a}, \boldsymbol{b}]$, and $\mathbf{H} = [\boldsymbol{y}, \boldsymbol{z}]^T$, with $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{y}$, and $\boldsymbol{z}$ being column vectors. An exact solution can also be obtained by finding $\mathbf{W} = [\boldsymbol{c}, \boldsymbol{d}, \boldsymbol{b}]$, and $\mathbf{H} = [\boldsymbol{y}, \boldsymbol{y}, \boldsymbol{z}]$, where $\boldsymbol{c}+\boldsymbol{d} = \boldsymbol{a}$, in a process known as factor splitting (Ulbrich et al. (2009)). Thus, a three factor solution introduces rotational ambiguity, as any two factors $\boldsymbol{c}$ and $\boldsymbol{d}$ can be chosen as long as they add up to $\boldsymbol{a}$. The same analysis can be seen by analyzing solutions with four or more factors, and we can conclude that a large amount of rotational ambiguity is a potential sign of overfitting.

1.7 Data

In this paper, we use the data from "Ambient Measurements of Highly Oxidized Gas-Phase Molecules during the Southern Oxidant and Aerosol Study (SOAS) 2013", measuring highly oxidized multifunctional molecules (HOMs) over a forest site in Alabama from June 22^{nd}, 2013 to July 7^{th} 2013 (Massoli et al. (2018)). The dataset contains mass spectra concentrations of 1,059 different ions over 27,336 different time stamps. Additionally, initial uncertainties associated with each measurement are also included. Positive Matrix Factorization was applied to the data using the PMF2 algorithm described in (Paatero (1997)), checking solutions from two to 10 factors, where a six factor solution was obtained (Massoli et al. (2018)). We note that some of the uncertainties were artificially increased for this PMF2 analysis. The authors concluded that a significant portion of the Secondary Organic Aerosol (SOA) was anthropogenic in origin (Massoli et al. (2018)).

In this paper, we use this six factor solution as a reference solution, and test whether the RHALS algorithm can recreate formulated factors as well as those found from PMF2. For reference, the PMF2 mass spectra factors and the total mass spectra concentrations over all of the data are shown in Figure 1. The time series factors, as well as the total time series concentrations, are shown in Figure 2. Both plots show total concentration amounts over the entire time series and mass spectra, respectively.

We will not discuss the chemical interpretations of the data, rather just the mathematical results from running the RHALS algorithm.

2 Methods

Here we present the derivation of the basic weighted HALS algorithm in Section 2.1, as well as a simple rotation algorithm in Section 2.2, our new external weighting algorithm in Section 2.3, and inclusion of L1 and L2 regularization in Section 2.4.

2.1 HALS Algorithm

The derivation of the unweighted algorithm for HALS is detailed in (Erichson et al. (2018)). Here we present a similar derivation, taking into account uncertainties in the data that act as weights.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

The cost function Q_j is defined by block coordinate descent, minimizing a "block", or outer product of individual factors, of $\mathbf{W}$ and $\mathbf{H}$ at a time.

$$\min Q_j = \min_{\mathbf{W}(:,j),\mathbf{H}(j,:)} ||(\mathbf{R}_j - \mathbf{W}(:,j)\mathbf{H}(j,:))/\mathbf{\Sigma}||_F^2 \tag{8}$$

In (8), $\mathbf{R}_j$ is the jth residual, with

$$\mathbf{R}_j = \mathbf{A} - \sum_{i \neq j}^{k} \mathbf{W}(:,i)\mathbf{H}(i,:) = \mathbf{A} - \mathbf{WH} + \mathbf{W}(:,j),\mathbf{H}(j,:) \tag{9}$$

$\mathbf{W}(:,j)$ is the jth column, or jth factor, of $\mathbf{W}$, $\mathbf{H}(j,:)$ is the jth row/factor of $\mathbf{H}$, and k is the number of factors that the algorithm is aiming to find. $\mathbf{\Sigma}$ contains the uncertainties associated with each element in the data matrix $\mathbf{A}$, and the symbol / is element-wise division.

To derive update rules for HALS, partial derivatives of Eq. (8) are taken with respect to the factors $\mathbf{W}(:,j)$ and $\mathbf{H}(j,:)$. With $\mathbf{\Sigma}$ present, this can become tricky, so we present a variation on the derivation presented in (Erichson et al. (2018)) by considering just a row and column of the weighted residual

$$Q_j^i = ||\mathbf{R}_j(i,:) - \mathbf{W}(i,j)\mathbf{H}(j,:))\mathbf{\Sigma}_i^{-1}||_F^2 \tag{10}$$

$$Q_j^p = ||\mathbf{\Sigma}_p^{-1}(\mathbf{R}_j(:,p) - \mathbf{H}(j,p)\mathbf{W}(:,j))||_F^2 \tag{11}$$

In Eq. (10) and Eq. (11), $\mathbf{\Sigma}_i$ are $\mathbf{\Sigma}_p$ are diagonal matrices with the diagonal elements corresponding to the elements of the ith row and pth column of $\mathbf{\Sigma}$, respectively. Expanding Eq. (10) and Eq. (11) using the fact that $||\mathbf{X}||_F^2 = Tr(\mathbf{X}^T\mathbf{X})$, where Tr takes the trace of the matrix, we get

$$Q_j^i = Tr(\mathbf{\Sigma}_i^{-1}\mathbf{R}_j^T(i,:)\mathbf{R}_j(i,:)\mathbf{\Sigma}_i^{-1} - 2\mathbf{W}(i,j)\mathbf{\Sigma}_i^{-1}\mathbf{R}_j^T(i,:)\mathbf{H}(j,:)\mathbf{\Sigma}_i^{-1} + \mathbf{W}(i,j)^2\mathbf{\Sigma}_i^{-1}\mathbf{H}^T(j,:)\mathbf{H}(j,:)\mathbf{\Sigma}_i^{-1}) \tag{12}$$

$$Q_j^p = Tr(\mathbf{R}_j^T(:,p)\mathbf{\Sigma}_p^{-1}\mathbf{\Sigma}_p^{-1}\mathbf{R}_j(:,p) - 2\mathbf{H}(j,p)\mathbf{R}_j^T(:,p)\mathbf{\Sigma}_p^{-1}\mathbf{\Sigma}_p^{-1}\mathbf{W}(:,j) + \mathbf{H}(j,p)^2\mathbf{W}^T(:,j)\mathbf{\Sigma}_p^{-1}\mathbf{\Sigma}_p^{-1}\mathbf{W}(:,j)) \tag{13}$$

Taking a derivative with respect to $\mathbf{W}(i,j)$ and $\mathbf{H}(j,p)$,

$$\frac{\partial Q_j^i}{\partial \mathbf{W}(i,j)} = -2Tr(\mathbf{\Sigma}_i^{-1}\mathbf{R}_j^T(i,:)\mathbf{H}(j,:)\mathbf{\Sigma}_i^{-1}) + 2\mathbf{W}(i,j)Tr(\mathbf{\Sigma}_i^{-1}\mathbf{H}^T(j,:)\mathbf{H}(j,:)\mathbf{\Sigma}_i^{-1}) \tag{14}$$

$$\frac{\partial Q_j^p}{\partial \mathbf{H}(j,p)} = -2Tr(\mathbf{R}_j^T(:,p)\mathbf{\Sigma}_p^{-1}\mathbf{\Sigma}_p^{-1}\mathbf{W}(:,j)) + 2\mathbf{H}(j,p)Tr(\mathbf{W}^T(:,j)\mathbf{\Sigma}_p^{-1}\mathbf{\Sigma}_p^{-1}\mathbf{W}(:,j)) \tag{15}$$

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

To eliminate the matrix traces, it is easy to show that Eq. (14) and Eq. (15) can be rewritten as

$$\frac{\partial Q_j^i}{\partial \mathbf{W}(i,j)} = -2\mathbf{R}_j(i,:)(\mathbf{H}^T(j,:)/(\mathbf{\Sigma}_i^T \odot \mathbf{\Sigma}_i^T)) + 2\mathbf{W}(i,j)\mathbf{H}(j,:)(\mathbf{H}^T(j,:)/(\mathbf{\Sigma}_i^T \odot \mathbf{\Sigma}_i^T)) \tag{16}$$

$$\frac{\partial Q_j^p}{\partial \mathbf{H}(j,p)} = -2\mathbf{R}_j^T(:,p)(\mathbf{W}(:,j)/(\mathbf{\Sigma}_p \odot \mathbf{\Sigma}_p)) + 2\mathbf{H}(j,p)\mathbf{W}^T(:,j)(\mathbf{W}(:,j)/(\mathbf{\Sigma}_p \odot \mathbf{\Sigma}_p)) \tag{17}$$

where / again represents element-wise division and $\odot$ represents element-wise multiplication. Setting Eq. (16) and Eq. (17) to zero and solving for the factor values yields

$$\mathbf{W}(i,j) = \frac{\mathbf{R}_j(i,:)(\mathbf{H}^T(j,:)/(\mathbf{\Sigma}_i^T \odot \mathbf{\Sigma}_i^T))}{\mathbf{H}(j,:)(\mathbf{H}^T(j,:)/(\mathbf{\Sigma}_i^T \odot \mathbf{\Sigma}_i^T))} \tag{18}$$

$$\mathbf{H}(j,p) = \frac{\mathbf{R}_j^T(:,p)(\mathbf{W}(:,j)/(\mathbf{\Sigma}_p \odot \mathbf{\Sigma}_p))}{\mathbf{W}^T(:,j)(\mathbf{W}(:,j)/(\mathbf{\Sigma}_p \odot \mathbf{\Sigma}_p))} \tag{19}$$

Substituting Eq. (9) into Eq. (18) and Eq. (19) yields the update rules:

$$\mathbf{W}(i,j) \leftarrow \left[\mathbf{W}(i,j) + \frac{\mathbf{A}(i,:)(\mathbf{H}^T(j,:)/(\mathbf{\Sigma}_i^T \odot \mathbf{\Sigma}_i^T)) - \mathbf{W}(i,:)\mathbf{H}(\mathbf{H}^T(j,:)/(\mathbf{\Sigma}_i^T \odot \mathbf{\Sigma}_i^T))}{\mathbf{H}(j,:)(\mathbf{H}^T(j,:)/(\mathbf{\Sigma}_i^T \odot \mathbf{\Sigma}_i^T))}\right]_+ \tag{20}$$

$$\mathbf{H}(j,p) \leftarrow \left[\mathbf{H}(j,p) + \frac{\mathbf{A}^T(:,p)(\mathbf{W}(:,j)/(\mathbf{\Sigma}_p \odot \mathbf{\Sigma}_p)) - \mathbf{H}^T(:,p)\mathbf{W}^T(\mathbf{W}(:,j)/(\mathbf{\Sigma}_p \odot \mathbf{\Sigma}_p))}{\mathbf{W}^T(:,j)(\mathbf{W}(:,j)/(\mathbf{\Sigma}_p \odot \mathbf{\Sigma}_p))}\right]_+ \tag{21}$$

In practice, the authors in Erichson et al. (2018) utilize the following simplified form of Eq. (20) and Eq. (21):

$$\mathbf{W}(i,j) \leftarrow \left[\mathbf{W}(i,j) - \frac{\nabla_{\mathbf{W}_{ij}} Q_j^i}{\nabla^2_{\mathbf{W}_{ij}} Q_j^i}\right]_+ \tag{22}$$

$$\mathbf{H}(j,p) \leftarrow \left[\mathbf{H}(j,p) - \frac{\nabla_{\mathbf{H}_{jp}} Q_j^p}{\nabla^2_{\mathbf{H}_{jp}} Q_j^p}\right]_+ \tag{23}$$

Thus, one can add additional auxiliary functions to the cost function Q, such as regularization and rotation terms, and add them to the update rules based on the new gradient and Hessian values. Furthermore, writing the update rules as in Eq. (22) and Eq. (23) includes the calculation of the projected gradient. This can be used as a stopping condition criterion that avoids the computational costs of calculating other convergence statistics (Erichson et al. (2018)).

We note that when the uncertainties are equal to 1, $\mathbf{\Sigma}$ can be disregarded in Eq. (20) and Eq. (21), and the update rules are identical to those in (Erichson et al. (2018)).

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

2.2 Rotational Considerations

Suppose a factor is interpretable by a scientist running a PMF algorithm, but a particular component is lacking or overrepresented. The scientist might be interested in a rotated solution that includes the complete factor. As detailed in (Paatero and Hopke (2009)), for a specific factor value $\mathbf{W}_{ij}$ or $\mathbf{H}_{jp}$, an auxillary term can be added to the cost function to pull the component towards a set value $\mathbf{W}^*_{ij}$ or $\mathbf{H}^*_{ij}$. Defining

$$Q^{i,j}_{aux_W} = s(\mathbf{W}_{ij} - \mathbf{W}^*_{ij})^2 \text{ or } Q^{p,j}_{aux_H} = s(\mathbf{H}_{jp} - \mathbf{H}^*_{jp})^2 \tag{24}$$

with s determining the strength of the pull, we find $\nabla_{\mathbf{W}_{ij}} Q^{i,j}_{aux_W} = 2s(\mathbf{W}_{ij} - \mathbf{W}^*_{ij})$, $\nabla^2_{\mathbf{W}_{ij}} Q^{i,j}_{aux_W} = 2s$, $\nabla_{\mathbf{H}_{j,p}} Q^{p,j}_{aux_H} = 2s(\mathbf{H}_{jp} - \mathbf{H}^*_{jp})$, and $\nabla^2_{\mathbf{H}_{jp}} Q^{j,p}_{aux_H} = 2s$. Adding these terms to the gradient and Hessian in Eq. (22) and Eq. (23), we can derive new update rules for $\mathbf{W}_{ij}$ and $\mathbf{H}_{jp}$

To probe possible rotations of entire factors, we introduce "pulling equations" to the cost function, which pull the elements of $\mathbf{W}$ and $\mathbf{H}$ to the desired rotations:

$$Q^{(j)}_{aux_H} = a(\sum_{i=1}^{n}(1 - \mathbf{H}_{ji}))^2 \tag{25}$$

$$Q^{(j)}_{aux_W} = b(\sum_{i=1}^{m}\mathbf{W}_{ij})^2 \tag{26}$$

These equations correspond to $r < 0$, with both a and b positive. For $r > 0$, $\mathbf{W}$ and $\mathbf{H}$ would be interchanged. These pulling equations are similar to those introduced in Paatero et al in (Paatero et al. (2002)), although their meanings are slightly changed. Simply, we aim to pull values of $\mathbf{H}$ towards more positive values and values of $\mathbf{W}$ towards 0, which mirrors the effect of setting r to be less than zero in Eq. (6).

Taking derivatives of Eq. (25) and Eq. (26),

$$\nabla_{\mathbf{W}_{ij}} Q^{j,i}_{aux_W} = b\mathbf{W}_{ij} \tag{27}$$

$$\nabla^2_{\mathbf{W}_{ij}} Q^{j,i}_{aux_W} = b \tag{28}$$

$$\nabla_{\mathbf{H}_{jp}} Q^{j,p}_{aux_H} = -a(1 - \mathbf{H}_{jp}) \tag{29}$$

$$\nabla^2_{\mathbf{H}_{jp}} Q^{j,p}_{aux_H} = a \tag{30}$$

we add these terms to the gradient and Hessian in Eq. (22) and Eq. (23) to receive new update rules.

Optimally, we would expect these equations to pull the values of $\mathbf{H}$ up and the values of $\mathbf{W}$ towards zero, and to reach different, potentially better solutions.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

2.3 External Weighting

To perform external weighting, we first find $\hat{\mathbf{A}} = \mathbf{A}./\mathbf{\Sigma}$, i.e., we divide the data elementwise by the uncertainties. RHALS is then applied to $\hat{\mathbf{A}}$ to form the estimated scaled factor matrices $\hat{\mathbf{W}}$ and $\hat{\mathbf{H}}$. The unscaled estimates $\mathbf{W}$ and $\mathbf{H}$ are found by the relation

$$\mathbf{WH} = (\hat{\mathbf{W}}\hat{\mathbf{H}}) \odot \Sigma \tag{31}$$

where $\odot$ represents elementwise multiplication. $\mathbf{W}$ and $\mathbf{H}$ can then be found iteratively via alternating least squares:

$$\mathbf{W} = ((\hat{\mathbf{W}}\hat{\mathbf{H}}) \odot \mathbf{\Sigma})\mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1} \text{ or } \mathbf{W} = ((\hat{\mathbf{W}}\hat{\mathbf{H}}) \odot \mathbf{\Sigma})\mathbf{H}^\dagger \tag{32}$$

$$\mathbf{H} = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T((\hat{\mathbf{W}}\hat{\mathbf{H}}) \odot \mathbf{\Sigma}) \text{ or } \mathbf{H} = \mathbf{W}^\dagger((\hat{\mathbf{W}}\hat{\mathbf{H}}) \odot \mathbf{\Sigma}) \tag{33}$$

Here, $\mathbf{W}^\dagger$ denotes the pseudoinverses of $\mathbf{W}$, and all negative elements are set to zero after the factor matrix is updated. Mathematically, these methods are identical, as long as the rank of the factor matrices is equal to k, and a short proof of this is detailed in section B of the Appendix. Of course, if the factor matrices were of lower rank (or contained very small singular values), then a lower rank factorization should first be found instead. Since both algorithms are identical mathematically, one could be favorable if it provided a speed advantage, depending on the computational efficiency of the pseudoinverse algorithm called. Running this code in Matlab on a single CPU, we found the update rules using the pseudoinverses were less computationally expensive.

To begin the iteration, we have to initiate either $\mathbf{W}$ or $\mathbf{H}$. The most intuitive way to do this is by assuming the weighted factors are similar to the unweighted factors, and setting $\mathbf{H}$ (or $\mathbf{W}$) to the weighted values, but scaled to the magnitude of the original data. Since each entry of the data matrix is approximated as $\mathbf{A}_{ij} = \sum_{p=1}^{k} \mathbf{W}_{ip}\mathbf{H}_{pj}$, and assuming relatively equal magnitudes for $\mathbf{W}$ and $\mathbf{H}$, the appropriate scale factor should be $\sqrt{\frac{mean(\mathbf{A})}{k}}$. When $\mathbf{W}$ and $\mathbf{H}$ contain values with greatly differing magnitudes, perhaps a different scaling factor should be used, although this has not been extensively tested. Thus, the algorithm should be initialized by

$$\mathbf{H}_0 = \sqrt{\frac{mean(\mathbf{A})}{k}} \frac{\mathbf{H}}{mean(\mathbf{H})} \tag{34}$$

Measuring the change in W and H between iterations, the algorithm typically converges relatively quickly - within 20 to 40 iterations. However, the algorithm may take longer for a larger number of factors and in certain cases. More work is needed regarding the different convergence issues that may arise.

2.4 Regularization

As in the RHALS algorithm presented in (Erichson et al. (2018)), we add L1 and L2 regularization. L1 regularization is added to the cost function through the L1 norm, $||||_1$, which is the sum of the absolute value of the components in a given row/column

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

in the factor matrices. L2 regularization is added through the L2, or Euclidean, norm, $||||_2^2$. L1 regularization is typically added to reduce sparsity, while L2 regularization is added to control the norms of the factors as well as avoid overfitting (Erichson et al. (2018)). The cost function now becomes

$$||(\mathbf{R}_j - \mathbf{W}(:,j)\mathbf{H}(j,:))/\mathbf{\Sigma}||_F^2 + \alpha||\mathbf{W}(:,j)||_1 + \beta||\mathbf{H}(j,:)||_1 + \gamma||\mathbf{W}(:,j)||_2^2 + \delta||\mathbf{H}(j,:)||_2^2 \quad (35)$$

In 35, α, β, γ, and δ control the amount of regularization that is added. Typically α and β, as well as γ and δ, are set equal. $\nabla_{\mathbf{W}_{ij}} Q_j^i = \beta + 2\gamma\mathbf{W}_{ij}$, $\nabla^2_{\mathbf{W}_{ij}} Q_j^i = 2\gamma$, and $\nabla_{\mathbf{H}_{jp}} Q_j^p = \alpha + 2\delta\mathbf{H}_{jp}$, $\nabla^2_{\mathbf{H}_{jp}} Q_j^p = 2\delta$. These terms are added to the update rules in Eq. (22) and Eq. (23).

What should α, β, γ, and δ in Eq. (35) be in practice? Optimal parameter values are often found using an L curve analysis, which measures the tradeoff between minimizing a norm and minimizing the residual error (Hansen and O'Leary (1993)).

In the SOAS dataset, the average data value is 1.9969×10^{-5}. When external weighting is applied, the average value of $\frac{\mathbf{A}}{\mathbf{\Sigma}}$ is 1.4851. L curves below are thus plotted using two different scales: one with a scale of 10^{-5}, and one with a scale of 1. The deterministic, unweighted algorithm is first applied to a small dataset consisting of the first 100 rows and columns of the SOAS dataset, scaled to contain different average values, to produce the L curves shown in Figure 3 Values of the regularization parameters are listed on the graphs, with "e" representing the exponential function with base 10. The optimal regularization value is one that is located in the bend of the graph, meaning it minimizes a solution's norm while having low residual error. In our results, we put preference on minimizing the latter, so we choose regularization values towards the left side of the bend. Analyzing the figures, the L1 regularization parameters α and β are chosen between 1×10^{-6} and 1×10^{-5} for the scale of 10^{-5}, and between 0.1 and 1 for the scale of 1. The L2 regularization parameters γ and δ are chosen around .001 for the scale of 10^{-5}, and 100 for the scale of 1. As a rule of thumb, the L1 regularization parameters should be chosen to be on the same order of magnitude as the data, and the L2 regularization parameters should be chosen to be around 100 times that. Note that neither the data matrix size nor the number of factors affect these optimal regularization parameter values. External and internal weighting will also not affect these values, as the magnitudes of the gradient and the Hessian will be around the same value.

3 Computational Details

The computer used for these calculations has an 11th Gen Intel® Core™ i7-1165G7 dual quad-core CPU with a speed of 2.80GHz. MATLAB™ R2021a is used for all calculations. While no algorithms detailed in this paper explicitly utilize multicore processing, they may rely on built in Matlab functions (such as svd()) that utilize multicore processing for increased efficiency.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

4 Results

We measure the accuracy of RHALS using the weighted residual error of the final algorithm again defined as $\sqrt{\left(\sum_i \sum_j \frac{\mathbf{A}_{ij} - \sum_k \mathbf{W}_{ik}\mathbf{H}_{kj}}{\sigma_{ij}}\right)^2}$, the correlation coefficients between the time series factors produced by RHALS and those of the reference solution, and the cosine similarity between the mass spectra profiles produced by RHALS and the reference solution.

The MU algorithm is initialized with random numbers, while the ALS and HALS algorithms are initialized by a non-negative SVD through an approach detailed in (Boutsidis and Gallopoulos (2008)). Note that the latter approach will still vary between random seeds, since the non-negative SVD is also found from utilizing the randomization technique in Section 1.3.1.

All algorithms cease updates when a stopping criterion is reached, or the maximum number of iterations is hit. For the MU and ALS algorithms, the value of the cost function is calculated after each iteration, and the algorithm halts when the percent change in this value, divided by the initial cost function value, is less than a set tolerance. The HALS algorithms use a different metric: the percent change in the projected gradient, which is detailed in (Lin (2007)). The projected gradient is defined as (Erichson et al. (2018))

$$\nabla^P_{\mathbf{W}_{ij}} \begin{cases} \nabla_{\mathbf{W}_{ij}} \text{ if } \mathbf{W}_{ij} > 0 \\ \min(0, \nabla_{\mathbf{W}_{ij}}) \text{ if } \mathbf{W}_{ij} = 0 \end{cases} . \tag{36}$$

and similarly for $\nabla^P_{\mathbf{H}_{jp}}$. When the percentage change in $\sum_i \sum_j (\nabla^P_{\mathbf{W}_{ij}})^2 + \sum_j \sum_p (\nabla^P_{\mathbf{H}_{jp}})^2$ (divided by its initial value) is less than a specified tolerance, the algorithm halts. Below, the maximum number of iterations allowed is set to be 100 (which is rarely reached by any algorithm), and the tolerance is set to be 1×10^{-4} for the HALS and ALS algorithms. A tolerance of 10^{-5} is used for the MU algorithm to account for a larger initial cost from the random initialization.

When comparing factors from different algorithms, it is important to note that the ratio of values between a time series factor and a mass spectra factor might be slightly different. Consider two algorithms that compute the identical factorizations

$$\mathbf{W}_1\mathbf{H}_1 = \sum_{i=1}^{k} \mathbf{W}_i^{(1)}\mathbf{H}_i^{(1)} \text{ and } \mathbf{W}_2\mathbf{H}_2 = \sum_{i=1}^{k} \mathbf{W}_i^{(2)}\mathbf{H}_i^{(2)} = \sum_{i=1}^{k} \left(c_i\mathbf{W}_i^{(1)}\right)\left(\frac{1}{c_i}\mathbf{H}^{(1)}\right)$$

with $\mathbf{W}_i$ being a column of $\mathbf{W}$, $\mathbf{H}_i$ being a row of $\mathbf{H}$, and c_k being an arbitrary positive constant. If we were to plot and compare $\mathbf{W}_i^{(1)}$ against $\mathbf{W}_i^{(2)}$, we would see that $\mathbf{W}_i^{(2)}$ is only c_i multiplied by the magnitude of $\mathbf{W}_i^{(1)}$, concluding one algorithm predicts a stronger signal for factor i, despite the fact that the factorizations are identical. Thus, when comparing individual factors between algorithms, the time series concentrations are cumulative over the entire mass spectra, and the mass spectra concentrations are cumulative over the entire time series (as also presented with the PMF2 results). Specifically, we plot the time series of factor i as the sum over the columns of the outer product $\mathbf{W}(:,i)\mathbf{H}(i,:)$, and we plot the mass spectra of factor i as the sum over the rows.

To avoid this issue, one can add scaling coefficients to rescale the factors at each iteration so that they will approximately have the same magnitude (Lu and Wu (2004)). This has not yet implemented in our code.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

4.1 Computational Efficiency

First, to show how the cost of RHALS scales with the size of the system, we consider the size of the data matrix vs the run time of the RHALS algorithm, as well as the deterministic internally weighted HALS and MU algorithms. Figure 4 shows how the computational costs of the different algorithms vary with the number of rows (and columns) in a square data matrix. The number of factors is set to five, and the data is formed by multiplying together two rank five matrices filled with sampled values from the PMF2 solution of the SOAS dataset. Uncertainty values are also randomly sampled from the given uncertainty data, and noise is added to the data matrix through random normal values that have a standard deviation of the uncertainties. L1 regularization is set to 1 for all factors, and L2 regularization is set to 50.

As one can see by looking at Figure 4, the logarithms of size and run time look to be proportional, and thus data size and run time look to be proportional with all three algorithms. RHALS clearly outperforms MU and HALS in terms of efficiency, taking mere seconds to run for each matrix size, about 1/20 of the time of the MU algorithm and 1/100 of the time of the deterministic HALS algorithm. It should also be noted that the randomization step, which accounts for about five percent of the algorithm computational costs, can be parallelized (Erichson et al. (2018)). Additionally, since the post algorithm external weighting step involves just matrix multiplication and an SVD, it can also be parallelized (Lahabar and Narayanan (2009)). This post weighting step accounts for around 60% of the algorithm's computational costs.

4.2 Simple Case

We now present the RHALS algorithm over a small test case. To do this, we will form a data matrix by combining underlying "true" factors and random noise. Thus, first we form a 5,952 by 400 matrix using the first three factors from the PMF2 solution to the SOAS dataset. We choose the time stamps between 9385 and 15336 from the PMF2 time series factor matrix $\mathbf{W}_{\text{PMF2}}$, since the PMF2 factors are clearly distinguishable from each other on this time interval. We choose the first 400 mass spectra profiles from the mass spectra factor matrix $\mathbf{H}_{\text{PMF2}}$, since the bulk of the mass spectra concentrations lie within this range.

A data matrix is then formed by combining the factors and adding random normal noise with a standard deviation equal to that of the uncertainty of the data:

$$\mathbf{A}_{test}(i,j) = \max(0, \mathbf{W}_{test}(i,:)\mathbf{H}_{test}(:,j) + N(0, \mathbf{\Sigma}(i,j))) \tag{37}$$

Next, we run RHALS over 20 trials, each with a different initialization, in order to obtain many different solutions that exist in the solution space. Here, initializations are formed using the randomized SVD method described earlier, which works well enough for generating different solutions. Figure 5 shows the weighted residual error of the RHALS algorithm over 20 trials.

The average weighted error over 20 trials is 1.4749×10^3, and the algorithm took an average time of 0.0561 seconds to converge with an average of 48.15 block coordinate descent steps per trial. As one can see, solutions can vary over different trials, and a smaller weighted error could be used as justification of one solution over another.

Next, we compare the similarities of the converged factors and the original factors that formed the test case. The averages of the similarities between the three factors are shown in Figure 6. A cosine similarity is used for the mass spectra, while the correlation coefficient is used for the time series.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

As one can see, the RHALS algorithm recreates the mass spectra and time series factors almost perfectly for all trials tested, regardless of the weighted residual error. In practice, a cosine similarity over .95 or a correlation coefficient over .90 between factor profiles represent almost identically interpretable solutions. Any trial is viable to be chosen as a "good" solution. Each solution yields a mass spectra similarity over 0.994, and a time series correlation over 0.974. There is low variance in the similarity metrics among trials, and higher weighted error thus corresponds with less important parts of the factor.

4.3 Large Dataset

4.3.1 Comparing Different Algorithms

Next, we analyze the complete dataset, and compare the RHALS factors to all PMF2 factors. Table 1 shows a table of diagnostics for the HALS, RHALS, ALS, and MU algorithms applied to the complete SOAS dataset, averaged over three trials. As one can see, algorithms with external weighting demonstrate a dramatic reduction in computational costs, albeit at a larger error. The ALS algorithms were by far the slowest, and the internally weighted HALS algorithm yielded the lowest average weighted error. The internally weighted MU algorithm produced comparable results to the internally weighted HALS algorithm in less than two thirds of the time; however, both the internally and externally weighted cases took the most steps to converge. The two fastest algorithms were the externally weighted HALS and RHALS algorithms, taking 1.01 and 0.50 seconds, respectively.

PMF factor profiles are often visually analyzed and compared to known candidate profiles for identification. Thus, the question arises of whether or not an algorithm utilizing external weighting still contains interpretable factors in light of the decreased accuracy. Figure 7 shows the cumulative time series from one solution of the externally and internally weighted HALS algorithm, with the factors produced by the externally weighted algorithm and the internally weighted algorithm overlaid. Figure 8 shows the cumulative mass spectra of this solution, with the factors from the externally weighted algorithm and the internally weighted algorithm laid out side by side. Similar graphs for the MU and ALS algorithms are detailed in the Appendix.

Upon visual inspection, the time series produced by the externally weighted factors have peaks and troughs at almost identical times, but the magnitudes of these peaks and troughs can vary throughout the factors. Interestingly, there exists a consistent difference in the magnitude of the time series between internal and external weighting. Specifically, the external weighting algorithms seem to consistently overpredict the concentrations of the time series of the second factor compared to the internally weighted algorthms. The difference is small compared to the concentrations of the factors, but this may lead to an over interpretation of the importance of the second factor with an externally weighted algorithm. It should be noted that there exists a (similarly) large variation in relative factor signals between trials of internally weighted algorithms, although cautious researchers should be aware of the possibility that external weighting introduces extra error in this analysis. More research is needed as to why the magnitude of some factors may be over or underestimated by external weighting.

Comparing the mass spectra yields a similar analysis. Most factors of externally weighted algorithms share the same spikes of key ions as in the internally weighted ions, although sometimes at different magnitudes. Occasionally, some externally weighted mass spectra factors will look quite unsimilar to the corresponding internally weighted factor, or contain noticeable spikes and divots in key ions, as seen in the second factor for the MU and HALS algorithm. It should also be noted that the

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

solutions of the externally weighted algorithms could be rotated away from each other. These differences may be extreme enough to encourage users to utilize multiple trials to search for multiple solutions and eliminate bad solutions.

4.3.2 Complete Analysis of RHALS Algorithm

Figure 9 shows the weighted residual error for a six factor solution (equivalent to the number of factors in PMF2) over 20 different non-negative SVD initializations. The average weighted residual error is 7.2743×10^3, with a convergence time of 0.5163 seconds over 38.45 steps per trial. If the solutions differ significantly, weighted residual error is a useful metric to pick a solution from 20 trials, and the 11^{th} and 13^{th} trials look like promising solutions, with weighted residual errors of 7.1474×10^3 and 7.1026×10^3, respectively.

The average similarities between RHALS factors and PMF2 factors are presented in FigSure 10, with the same similarity metrics as listed above.

Only the 2^{nd}, 11^{th}, 13^{th} and 17^{th} trials have an average cosine similarity between mass spectra factors over .95, along with a correlation over .90 between time series factors. The 13^{th} trial holds the highest similarity scores, with a time series correlation of 0.9468 and a cosine similarity of 0.9787. As seen in Figure 9, these solutions have a small weighted residual error, further justifying picking a solution with low weighted error.

How often does RHALS find a "good" solution? As we saw with the small test case, every RHALS solution contained factors very similar to the true factors. But only four out of twenty trials produced solutions surpassing 0.95 in mass spectra similarity and 0.90 in time series correlation for the entire dataset. Running 100 trials, it was found that RHALS found a solution matching these criteria 27% of the time. Thus the probability of not finding a good solution in 10 trials would be around $(1 - .27)^{10} \approx 4.3\%$, and $\approx 0.2\%$ in 20 trials. This rate could vary depending on the number of factors and the problem itself, and should be investigated further.

We now explore another way one could pick a solution out of 20 trials. Researchers might be interested in seeing which factor profiles show up the most. Thus a solution that contains factors that are repeatedly found might be preferable. Thus in Figure 11 we compare a given trial to all other trials by plotting the average similarity between that trial and the other trial. Again, cosine similarity is used for the mass spectra and the correlation coefficient is used for the time series. The bottom plot shows the two graphs averaged to give a total metric of the similarity of a solution to the other solutions.

Analyzing these figures, it appears that most solutions have about the same similarity to each other, with a time series correlation of around 0.87 and a mass spectra similarity of around 0.93. A few solutions, such as those in the trials between 6 and 9, can be ruled out as outliers, due to a low similarity to the other solutions. While the 12^{th} trial yields the highest similarity between it and the other solutions (0.9148 averaged between the time series correlation and the mass spectra similarity), no solution drastically outperforms many of the other solutions. Note that the solutions from the 2^{nd}, 11^{th}, 13^{th} and 17^{th} trials all perform well in this analysis, with average similarities of 0.9028, 0.8993, 0.9016, and 0.9036, respectively.

From visual inspection of figures 9 and 11, the solution from the 13^{th} trial is barely outperformed in the above analysis, while holding a clear lead in accuracy. Thus, it seems to be a rational choice to decide that this solution is the "best". The solution from the 13^{th} trial is plotted in Figures 12 and 13. Figure 12 shows the cumulative mass spectra profiles from the

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

RHALS solution, while Figure 13 shows the cumulative time series profiles from the RHALS solution compared with those from the PMF2 solution. Again, the profiles are plotted as a sum over the entire time series and mass spectra, respectively.

The similarities of the RHALS factors to the "true" PMF2 factors are very promising; however, the magnitude of some factors differ. Specifically, RHALS underestimates the significance of the first factor, while overestimating the significance of the second and fifth factors. Again, this additional error should be considered when interpreting RHALS factors.

Another potential concern from the RHALS algorithm is the factors having a general bias towards higher or lower magnitudes. Specifically, since RHALS enforces non negativity merely by setting negative elements to zero, in addition to regularization, it could be hypothesized that RHALS would produce factors of lower magnitude than the actual data. In Figure A1, the time series of the six RHALS factors, stacked on top of each other, are plotted against the total time series of the data matrix. Some times see larger magnitudes from the RHALS algorithm, along with other times seeing smaller magnitudes, but there seems to be no general pattern of bias towards solutions of greater or lesser magnitude generated by RHALS.

Finally, we determine if a solution with a different number of factors is more optimal with the RHALS algorithm for this dataset. Figure A6 shows the convergence of weighted residual error of RHALS as the number of factors increases, with the random initialization from the 13^{th} trial. Error dramatically decreases as the factors increase from one to six, while barely improving for solutions with a larger number of factors. Thus, a six factor solution is justified by the RHALS algorithm.

4.4 Testing Rotations

For rotations of entire factors, we first tested the rotational algorithm detailed in (Paatero and Hopke (2009)) to see whether it was applicable to RHALS. Unfortunately the pulling equations used in the Multilinear Engine (ME) do not seem to also work for RHALS.

Implementing the approach laid out in Section 2.2 to some of the solutions, we find that this method can potentially find better factorizations in the solution space. Ranging the values of a and b in Eq. (25) and Eq. (26) between 0 and 500 (which is of similar magnitude of the optimal amount of L2 regularization), we test the rotational method on the RHALS solutions from the 11^{th}, 13^{th} and 17^{th} trials. The graphs are plotted in which a positive pulling value corresponds to a "pull up" of **W** and a "pull down" of **H**, and vice versa for negative pulling values. The values of a and b are the magnitude of the pulling parameter. We present the rotation of the 13^{th} solution in Figures 14 and 15, with the rotations of the solutions from other trials in the Appendix.

As one can see by analyzing the rotation of the solution from the three different trials, solutions that have lower weighted residual error and are closer to the target solution can be found through this simple pulling method, regardless of the direction of the pull. Interestingly, weighted residual error was able to be decreased to under 7000 when rotating the solution from the 13^{th} trial towards larger values of W with a pulling parameter of 400. There also isn't always an increase in values for the factors that are being pulled up, as seen in Figure A11 in the Appendix. Sometimes, in fact, as the factors are being pulled one direction, as most clearly seen with the rotated factors from the 11^{th} trial, the matrix norms respond in the opposite direction.

The magnitude of change in the similarity is small for all solutions, and rotations may only be worthwhile to look at when no good solution exists from the onset. The benefit of the rotations is more apparent when rotating poor solutions.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

5 Conclusions

As the size of datasets has grown, computational costs have become increasingly expensive for traditional PMF algorithms. Thus, randomized and hierarchical algorithms are attractive alternative methods. Specifically, the RHALS algorithm was shown to provide a reduction in run time compared to the multiplicative update algorithm, as well as the deterministic HALS algorithm. Furthermore, we proposed a novel approach to handling uncertainties in a weighted factorization problem. While this approach, coupled with randomization, slightly reduced the accuracy of the algorithm, it dramatically decreased the computational cost. Ultimately, we showed that our weighted RHALS algorithm was able to almost completely recreate the factors in both a formed test matrix and a real dataset, and is a useful tool to finding non-negative factors of large datasets.

Code and data availability.

The source code for the algorithms detailed in this paper are given at https://zenodo.org/badge/latestdoi/537111580. The datasets used for this analysis are given at https://doi.org/10.25810/0829-t279.

Appendix A: ALS Derivation

The cost function Q is defined as

$$Q = ||\frac{\mathbf{A}-\mathbf{WH}}{\mathbf{\Sigma}}||_F^2 + \sum_{i=1}^{m}\alpha||\mathbf{W}(i,:)||_1 + \sum_{j=1}^{n}\beta||\mathbf{H}(:,j)||_1 + \gamma||\mathbf{W}||_F^2 + \delta||\mathbf{H}||_F^2 \tag{A1}$$

where α, β, γ, and δ are regularization parameters. As with the derivation of HALS, the elementwise division of $\mathbf{\Sigma}$ is eliminated by considering a row or column of the residual at a time. Thus, we find

$$0 \leftarrow \frac{\partial Q^i}{\partial \mathbf{W}(i,:)} = \frac{\partial}{\partial \mathbf{W}(i,:)}(||(\mathbf{A}(i,:) - \mathbf{W}(i,:)\mathbf{H})\mathbf{\Sigma}_i^{-1}||_F^2 + \alpha||\mathbf{W}(i,:)||_1 + \sum_{j=1}^{n}\beta||\mathbf{H}(:,j)||_1 + \gamma||\mathbf{W}(i,:)||_2^2 + \delta||\mathbf{H}||_F^2) \tag{A2}$$

$$0 \leftarrow \frac{\partial Q^j}{\partial \mathbf{H}(:,j)} = \frac{\partial}{\partial \mathbf{H}(:,j)}(||\mathbf{\Sigma}_j^{-1}(\mathbf{A}(:,j) - \mathbf{WH}(:,j))||_F^2 + \sum_{i=1}^{m}\alpha||\mathbf{W}(i,:)||_1 + \beta||\mathbf{H}(:,j)||_1 + \gamma||\mathbf{W}||_F^2 + \delta||\mathbf{H}(:,j)||_2^2) \tag{A3}$$

where $\mathbf{\Sigma}_i$ is a diagonal $n \times n$ matrix with the diagonal values equal to the ith row of $\mathbf{\Sigma}$, and $\mathbf{\Sigma}_j$ is a diagonal $m \times m$ matrix with the diagonal values equal to the jth column of $\mathbf{\Sigma}$. Using the fact that $||\mathbf{X}||_F^2 = Tr(\mathbf{X}^T\mathbf{X})$, with Tr being the trace of the matrix, Eq. (A2) and Eq. (A3) can be rewritten as

$$0 = \frac{\partial}{\partial \mathbf{W}_i}(Tr(\mathbf{\Sigma}_i^{-1}\mathbf{A}_i^T\mathbf{A}_i\mathbf{\Sigma}_i^{-1} - 2\mathbf{\Sigma}_i^{-1}\mathbf{A}_i^T\mathbf{W}_i\mathbf{H}\mathbf{\Sigma}_i^{-1} + \mathbf{\Sigma}_i^{-1}\mathbf{H}^T\mathbf{W}_i^T\mathbf{W}_i\mathbf{H}\mathbf{\Sigma}_i^{-1}) + \alpha||\mathbf{W}(i,:)||_1 + \sum_{j=1}^{n}\beta||\mathbf{H}(:,j)||_1 + \gamma||\mathbf{W}(i,:)||_2^2 + \delta||\mathbf{H}||_F^2) \tag{A4}$$

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

$$0 = \frac{\partial}{\partial \mathbf{H}_j}(Tr(\mathbf{A}_j^T\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\Sigma}_j^{-1}\mathbf{A}_j - 2\mathbf{H}_j^T\mathbf{W}^T\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\Sigma}_j^{-1}\mathbf{A}_j + \mathbf{H}_j^T\mathbf{W}^T\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\Sigma}_j^{-1}\mathbf{W}\mathbf{H}_j) + \sum_{i=1}^{m}\alpha||\mathbf{W}(i,:)||_1 + \beta||\mathbf{H}(:,j)||_1 + \gamma||\mathbf{W}||_F^2 + \delta||\mathbf{H}(:,j)||_2^2) \quad \text{(A5)}$$

where $\mathbf{A}(i,:) = \mathbf{A}_i$, $\mathbf{A}(:,j) = \mathbf{A}_j$, $\mathbf{W}(i,:) = \mathbf{W}_i$, and $\mathbf{H}(:,j) = \mathbf{H}_j$. Using vector derivative rules and the fact that $Tr(\mathbf{ABC}) = Tr(\mathbf{CAB}) = Tr(\mathbf{BCA})$ when $\mathbf{ABC}$ is square, the following gradients are found.

$$0 \leftarrow -2\mathbf{A}_i\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\Sigma}_i^{-1}\mathbf{H}^T + 2\mathbf{W}_i\mathbf{H}\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\Sigma}_i^{-1}\mathbf{H}^T + \alpha\overrightarrow{1}_{1\times k} + 2\gamma\mathbf{W}_i \quad \text{(A6)}$$

$$0 \leftarrow -2\mathbf{W}^T\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\Sigma}_j^{-1}\mathbf{A}_j + 2\mathbf{W}^T\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\Sigma}_j^{-1}\mathbf{W}\mathbf{H}_j + \beta\overrightarrow{1}_{k\times 1} + 2\delta\mathbf{H}_j \quad \text{(A7)}$$

In Eq. (A6) and Eq. (A7), $\overrightarrow{1}$ is a vector of ones. Finally, the following update rules are found.

$$\mathbf{W}_i \leftarrow [(\mathbf{A}_i\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\Sigma}_i^{-1}\mathbf{H}^T - \frac{\alpha}{2}\overrightarrow{1}_{1\times k})(\mathbf{H}\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\Sigma}_i^{-1}\mathbf{H}^T + \gamma\mathbf{I}_{k\times k})^{-1}]_+ \quad \text{(A8)}$$

$$\mathbf{H}_j \leftarrow [(\mathbf{W}^T\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\Sigma}_j^{-1}\mathbf{W} + \delta\mathbf{I}_{k\times k})^{-1}(\mathbf{W}^T\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\Sigma}_j^{-1}\mathbf{A}_j - \frac{\beta}{2}\overrightarrow{1}_{k\times 1})]_+ \quad \text{(A9)}$$

where $\mathbf{I}$ is the identity matrix and $[\boldsymbol{v}]_+$ returns all negative entries in $\boldsymbol{v}$ as 0.

Appendix B: Mathematical Equivalence of Pseudoinverse Update and Ordinary Least Squares

For simplicity, we label $(\tilde{\mathbf{W}}\tilde{\mathbf{H}}) \odot \boldsymbol{\Sigma}$ as $\tilde{\mathbf{y}}$. Let's consider the ordinary least squares update for W:

$$\mathbf{W} = \tilde{\mathbf{y}}\mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}$$

$\mathbf{H}$ is a $k \times n$ matrix that is assumed to be of full column rank, and thus can be decomposed into the exact SVD factorization $\mathbf{H} = \mathbf{USV}^T$, where $\mathbf{U}$ is a $k \times k$ orthogonal matrix, $\mathbf{S}$ is a $k \times k$ diagonal matrix, and $\mathbf{V}$ is a $n \times k$ matrix with orthonormal columns. The pseudoinverse of $\mathbf{H}$ is defined as $\mathbf{VS}^{-1}\mathbf{U}^T$. $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, and $\mathbf{U}^T = \mathbf{U}^{-1}$. The update for $\mathbf{W}$ can be rewritten as

$$\begin{aligned}
\mathbf{W} &= \tilde{\mathbf{y}}\mathbf{VSU}^T(\mathbf{USISU}^T)^{-1} \\
&= \tilde{\mathbf{y}}\mathbf{VSU}^T(\mathbf{US}^2\mathbf{U}^T)^{-1} \\
&= \tilde{\mathbf{y}}\mathbf{VSU}^T(\mathbf{US}^{-2}\mathbf{U}^T) \\
&= \tilde{\mathbf{y}}\mathbf{VSIS}^{-2}\mathbf{U}^T \\
&= \tilde{\mathbf{y}}(\mathbf{VS}^{-1}\mathbf{U}^T) \\
&= \tilde{\mathbf{y}}\mathbf{H}^\dagger \quad \text{(A1)}
\end{aligned}$$

which is equal to the pseudoinverse update described in Section 2.3. A similar argument can be made for the update for $\mathbf{H}$.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

Author contributions.

B.C.S. wrote the manuscript, conducted all of the numerical experiments, and derived the equations. D.K.H. advised the numerical and theoretical approach. D.K.H., M. C., and H.S. reviewed and edited the manuscript. M.C. and H.S. provided the AMS data set and reference PMF2 solution.

Competing interests.

The authors declare that they have no conflict of interest.

Acknowledgements. B.C.S. acknowledges funding from CU SPUR (Summer Program for Undergraduate Research), D.K.H. acknowledges support from NASA 80NSSC20K0214.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

References

Boutsidis, C. and Gallopoulos, E.: SVD based initialization: A head start for nonnegative matrix factorization, Pattern Recognition, 41, 1350–1362, https://doi.org/10.1016/j.patcog.2007.09.010, 2008.

Burred, J. J.: Detailed derivation of multiplicative update rules for NMF, p. 8.

Cichocki, A. and Phan, A.-H.: Fast Local Algorithms for Large Scale Nonnegative Matrix and Tensor Factorizations, IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, E92-A, 708–721, https://search.ieice.org/bin/summary.php?id=e92-a_3_708&category=A&year=2009&lang=E&abst=, publisher: The Institute of Electronics, Information and Communication Engineers, 2009.

Eckart, C. and Young, G.: The approximation of one matrix by another of lower rank, Psychometrika, 1, 211–218, https://doi.org/10.1007/BF02288367, 1936.

Erichson, N. B., Mendible, A., Wihlborn, S., and Kutz, J. N.: Randomized Nonnegative Matrix Factorization, Pattern Recognition Letters, 104, 1–7, https://doi.org/10.1016/j.patrec.2018.01.007, arXiv:1711.02037 [cs, stat], 2018.

Halko, N., Martinsson, P. G., and Tropp, J. A.: Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions, SIAM Review, 53, 217–288, https://doi.org/10.1137/090771806, publisher: Society for Industrial and Applied Mathematics, 2011.

Hansen, P. C. and O'Leary, D. P.: The Use of the L-Curve in the Regularization of Discrete Ill-Posed Problems, SIAM Journal on Scientific Computing, 14, 1487–1503, https://doi.org/10.1137/0914086, publisher: Society for Industrial and Applied Mathematics, 1993.

Kaloorazi, M. F. and Chen, J.: Randomized Truncated Pivoted QLP Factorization for Low-Rank Matrix Recovery, IEEE Signal Processing Letters, 26, 1075–1079, https://doi.org/10.1109/LSP.2019.2920054, conference Name: IEEE Signal Processing Letters, 2019.

Kim, E. and Hopke, P. K.: Source Identifications of Airborne Fine Particles Using Positive Matrix Factorization and U.S. Environmental Protection Agency Positive Matrix Factorization, Journal of the Air & Waste Management Association, 57, 811–819, https://doi.org/10.3155/1047-3289.57.7.811, publisher: Taylor & Francis _eprint: https://doi.org/10.3155/1047-3289.57.7.811, 2007.

Kumar, K.: Principal component analysis: Most favourite tool in chemometrics, Resonance, 22, 747–759, https://doi.org/10.1007/s12045-017-0523-9, 2017.

Lahabar, S. and Narayanan, P. J.: Singular value decomposition on GPU using CUDA, in: 2009 IEEE International Symposium on Parallel & Distributed Processing, pp. 1–10, https://doi.org/10.1109/IPDPS.2009.5161058, iSSN: 1530-2075, 2009.

Lee, D. D. and Seung, H. S.: Learning the parts of objects by non-negative matrix factorization, Nature, 401, 788–791, https://doi.org/10.1038/44565, number: 6755 Publisher: Nature Publishing Group, 1999.

Lin, C.-J.: Projected Gradient Methods for Nonnegative Matrix Factorization, Neural Computation, 19, 2756–2779, https://doi.org/10.1162/neco.2007.19.10.2756, conference Name: Neural Computation, 2007.

Lu, J. and Wu, L.: Technical details and programming guide for a general two-way positive matrix factorization algorithm, Journal of Chemometrics, 18, 519–525, https://doi.org/10.1002/cem.894, _eprint: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/pdf/10.1002/cem.894, 2004.

Massoli, P., Stark, H., Canagaratna, M. R., Krechmer, J. E., Xu, L., Ng, N. L., Mauldin, R. L., Yan, C., Kimmel, J., Misztal, P. K., Jimenez, J. L., Jayne, J. T., and Worsnop, D. R.: Ambient Measurements of Highly Oxidized Gas-Phase Molecules during the Southern Oxidant and Aerosol Study (SOAS) 2013, ACS Earth and Space Chemistry, 2, 653–672, https://doi.org/10.1021/acsearthspacechem.8b00028, publisher: American Chemical Society, 2018.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

Nasre, R., Burtscher, M., and Pingali, K.: Morph algorithms on GPUs, in: Proceedings of the 18th ACM SIGPLAN symposium on Principles and practice of parallel programming, PPoPP '13, pp. 147–156, Association for Computing Machinery, New York, NY, USA, https://doi.org/10.1145/2442516.2442531, 2013.

Paatero, P.: Least squares formulation of robust non-negative factor analysis, Chemometrics and Intelligent Laboratory Systems, 37, 23–35, https://doi.org/10.1016/S0169-7439(96)00044-5, 1997.

Paatero, P. and Hopke, P. K.: Rotational tools for factor analytic models, Journal of Chemometrics, 23, 91–100, https://doi.org/10.1002/cem.1197, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.1197, 2009.

Paatero, P. and Tapper, U.: Analysis of different modes of factor analysis as least squares fit problems, Chemometrics and Intelligent Laboratory Systems, 18, 183–194, https://doi.org/10.1016/0169-7439(93)80055-M, 1993.

Paatero, P. and Tapper, U.: Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values, Environmetrics, 5, 111–126, https://doi.org/10.1002/env.3170050203, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/env.3170050203, 1994.

Paatero, P., Hopke, P. K., Song, X.-H., and Ramadan, Z.: Understanding and controlling rotations in factor analytic models, Chemometrics and Intelligent Laboratory Systems, 60, 253–264, https://doi.org/10.1016/S0169-7439(01)00200-3, 2002.

Paatero, P., Hopke, P. K., Begum, B. A., and Biswas, S. K.: A graphical diagnostic method for assessing the rotation in factor analytical models of atmospheric pollution, Atmospheric Environment, 39, 193–201, https://doi.org/10.1016/j.atmosenv.2004.08.018, 2005.

Takács, G. and Tikk, D.: Alternating least squares for personalized ranking, in: Proceedings of the sixth ACM conference on Recommender systems, RecSys '12, pp. 83–90, Association for Computing Machinery, New York, NY, USA, https://doi.org/10.1145/2365952.2365972, 2012.

Tan, W., C. S. F. L. L. C. W. Z. and Cao, L.: Matrix Factorization on GPUs with Memory Optimization and Approximate Computing, 26.

Ulbrich, I. M., Canagaratna, M. R., Zhang, Q., Worsnop, D. R., and Jimenez, J. L.: Interpretation of organic components from Positive Matrix Factorization of aerosol mass spectrometric data, Atmospheric Chemistry and Physics, 9, 2891–2918, https://doi.org/10.5194/acp-9-2891-2009, publisher: Copernicus GmbH, 2009.

Wei, S., Zheng, X., Chen, D., and Chen, C.: A hybrid approach for movie recommendation via tags and ratings, Electronic Commerce Research and Applications, 18, 83–94, https://doi.org/10.1016/j.elerap.2016.01.003, 2016.

Xie, Y.-L., Hopke, P. K., and Paatero, P.: Positive matrix factorization applied to a curve resolution problem, Journal of Chemometrics, 12, 357–364, https://doi.org/10.1002/(SICI)1099-128X(199811/12)12:6<357::AID-CEM523>3.0.CO;2-S, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291099-128X%28199811/12%2912%3A6%3C357%3A%3AAID-CEM523%3E3.0.CO%3B2-S, 1998.

Yahaya, F., Puigt, M., Delmaire, G., and Roussel, G.: How to Apply Random Projections to Nonnegative Matrix Factorization with Missing Entries?, in: 2019 27th European Signal Processing Conference (EUSIPCO), pp. 1–5, https://doi.org/10.23919/EUSIPCO.2019.8903036, iSSN: 2076-1465, 2019.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

Table 1. Average statistics of different algorithms over 5 different SVD initializations, tolerance=!0^{-4} (for MU, tolerance is 10^{-5} and a random initialization), L1 Regularization=1, L2 Regularization=50.

Comparison of Algorithms			
Algorithm	Total Time (s)	Steps	Weighted Error
ALS (IW)	317.08	42.2	6.48×10^3
ALS (EW)	12.21	29.0	7.35×10^3
MU (IW)	19.31	76.4	6.52×10^3
MU (EW)	4.06	51.2	7.11×10^3
MU (Randomized)	4.02	53.6	7.76×10^3
HALS (IW)	33.87	19.4	6.45×10^3
HALS (EW)	1.01	31.2	7.08×10^3
RHALS	0.50	36.6	7.15×10^3

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

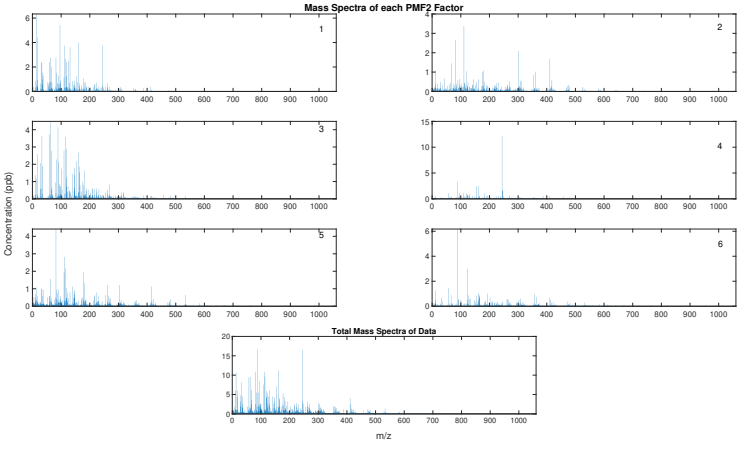

Figure 1. Mass spectra profiles of the 6 PMF2 factors labeled by order, as well as the total mass spectra concentration of the data.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

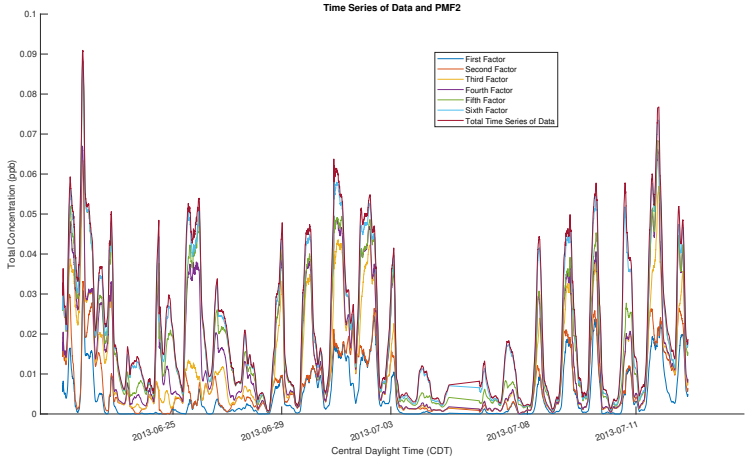

Figure 2. Time series profiles of the six PMF2 factors, overlaid with the total time series concentration of the data. A rolling average is used with values representing average concentrations over the previous two hours. The individual time series are stacked on top of each other in order to compare the total PMF2 time series to the overall time series of the data.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

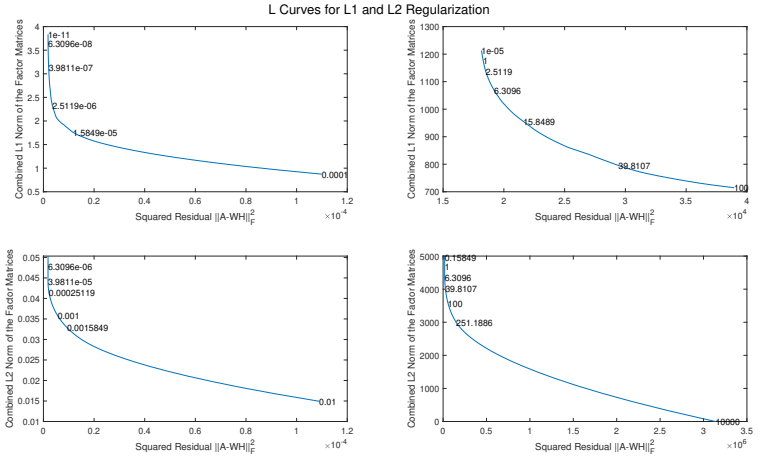

Figure 3. L Curves for L1 and L2 Regularization, applied to data with different average values. The top left graph shows L1 regularization for data with a magnitude around 10^{-5}; the top right graph shows L1 regularization for data with a magnitude around 1; the bottom left graph shows L2 regularization for data with a magnitude around 10^{-5}; the bottom right graph shows L2 regularization for data with a magnitude around 1.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

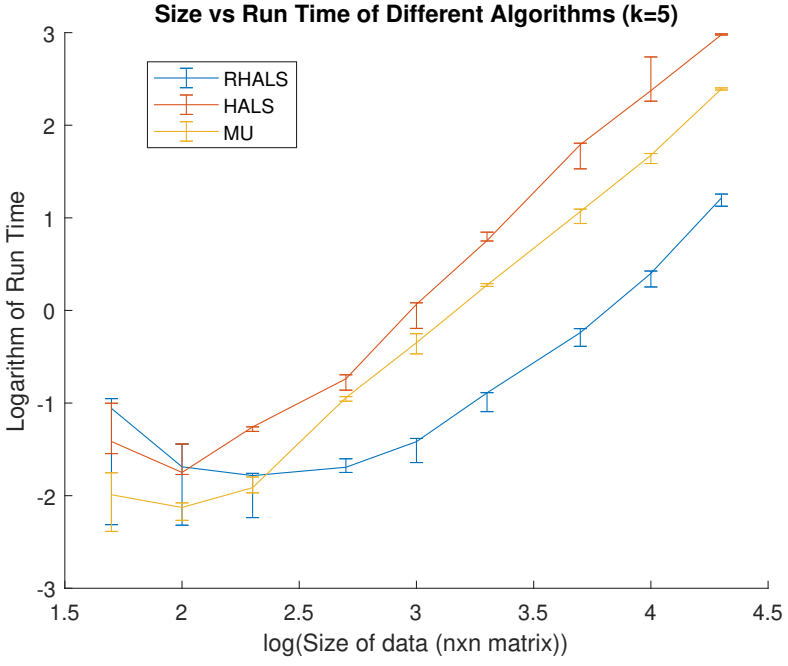

Figure 4. Size of data matrix vs run time (in seconds) in RHALS algorithm, performed with three different random seeds. The median run time is plotted, with the error bars plotting the maximum and minimum run times. Note that run time initially decreases due to a coincidental reduction in the number of steps to convergence.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

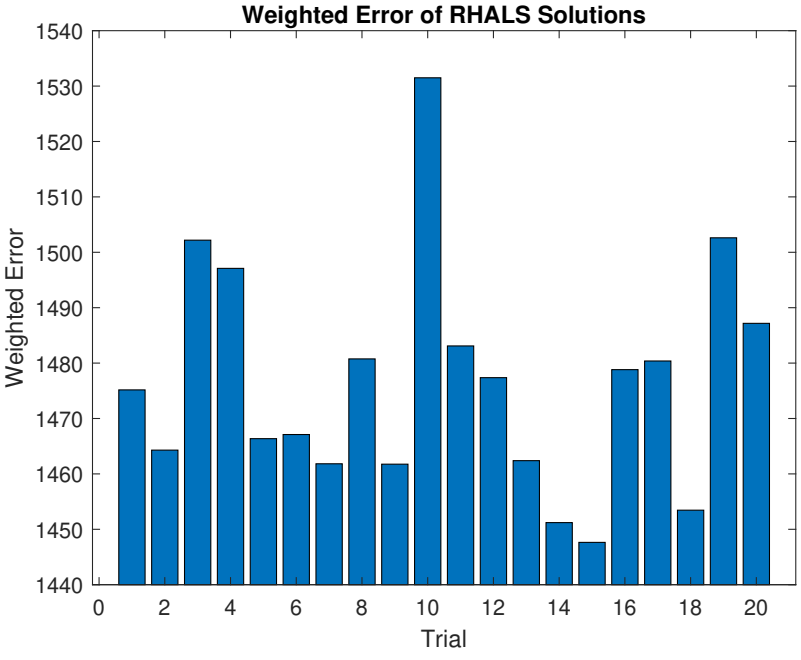

Figure 5. RHALS error over 20 trials. Mean: 1476.6, Standard Deviation: 20.4740.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

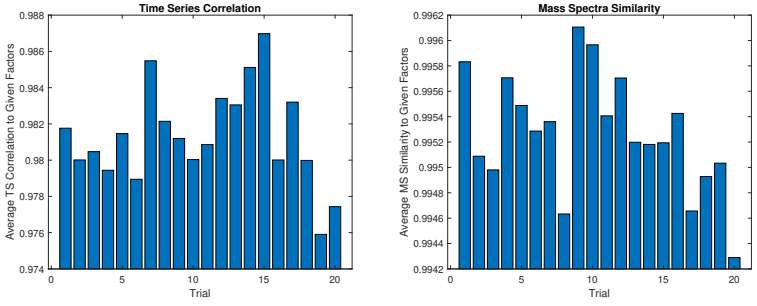

Figure 6. Average mass spectra and time series similarity to formed factors in the small test case.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

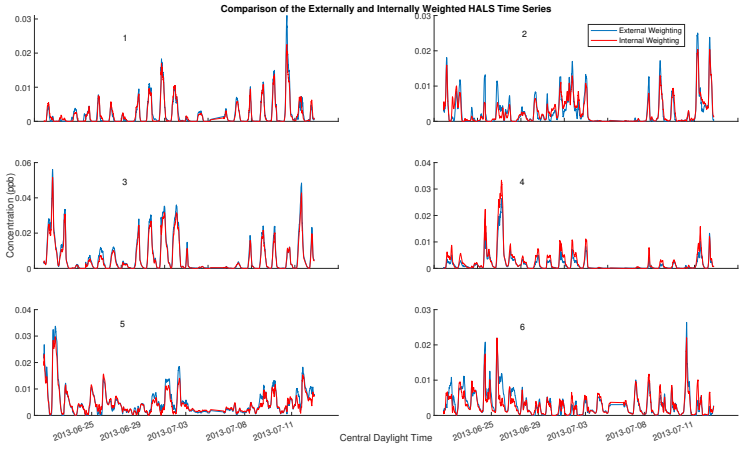

Figure 7. Comparison of two hour rolling average of time series for externally Weighted and internally weighted HALS factors. Externally weighted error: 7.1120×10^3. Internally weighted error: 6.4657×10^3.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

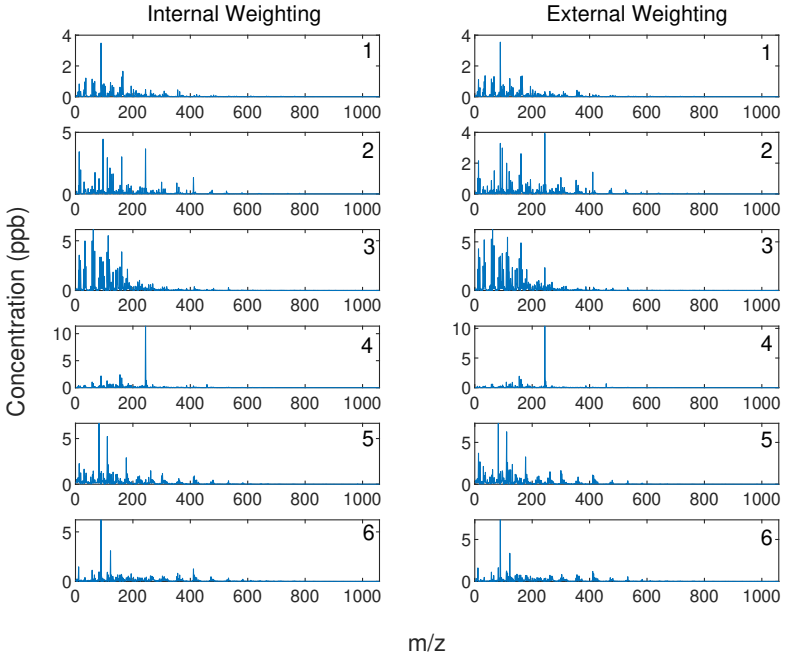

Figure 8. Comparison of mass spectra for externally weighted and internally weighted HALS factors.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

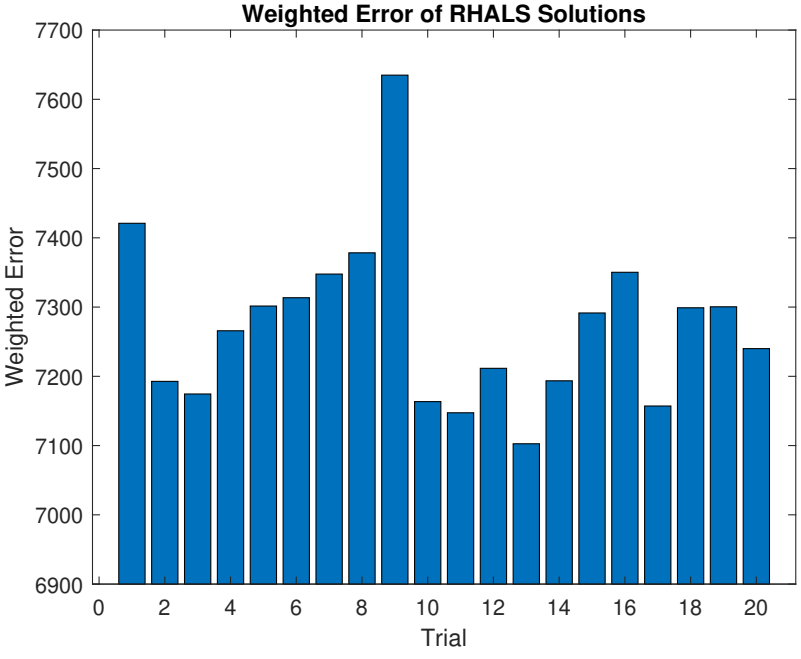

Figure 9. RHALS error over 20 trials. Mean Error: 7.2743×10^3, Standard Deviation of Error: 120.81. Mean Time: 0.5163, Standard Deviation of Time: 0.0292. Mean Steps: 38.45 Standard Deviation of Steps: 6.8785.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

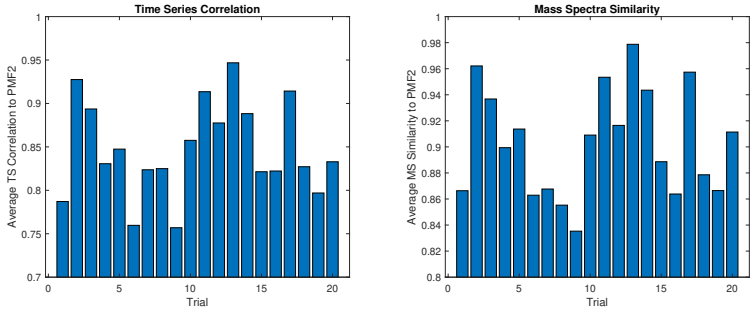

Figure 10. Average RHALS mass spectra and time series similarities to PMF2.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

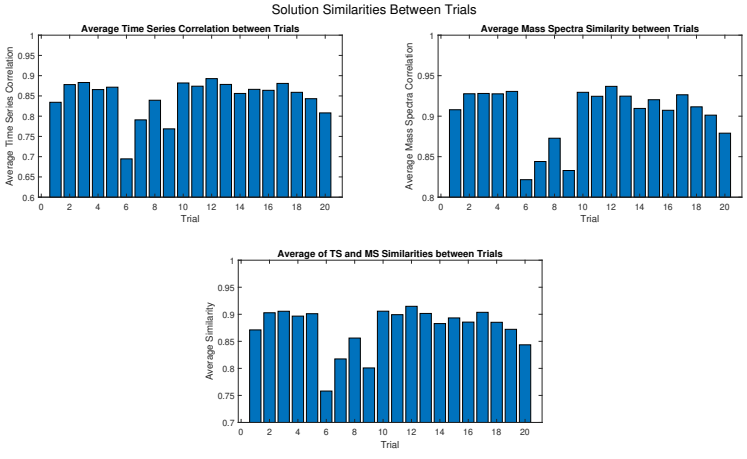

Figure 11. Similarity between the solutions of the different trials for RHALS

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

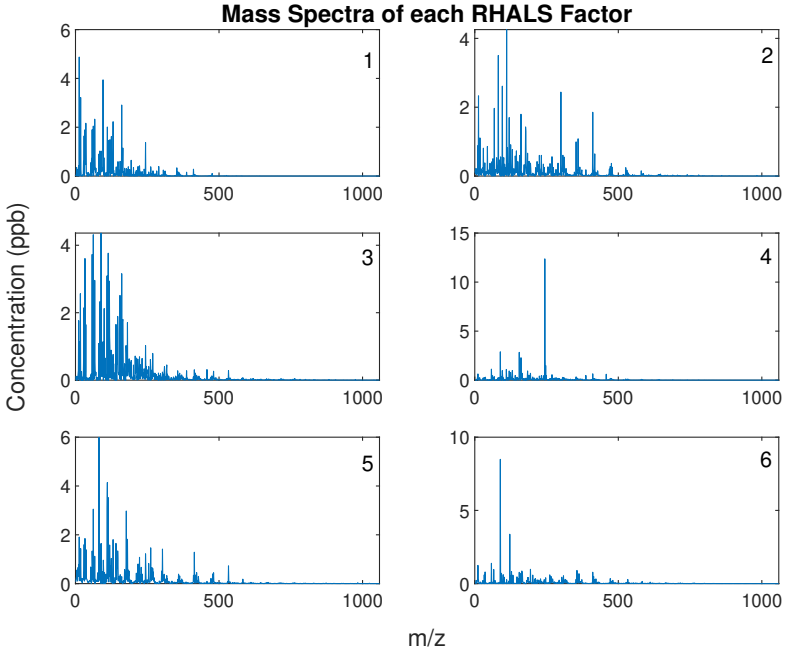

Figure 12. RHALS Mass Spectra Series Factors, labeled by order.; First factor similarity with PMF2=0.9631. Second factor similarity with PMF2=0.9582. Third factor similarity with PMF2=0.9912. Fourth factor similarity with PMF2=0.9970. Fifth factor similarity with PMF2=0.9795. Sixth factor similarity with PMF2=0.9835.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

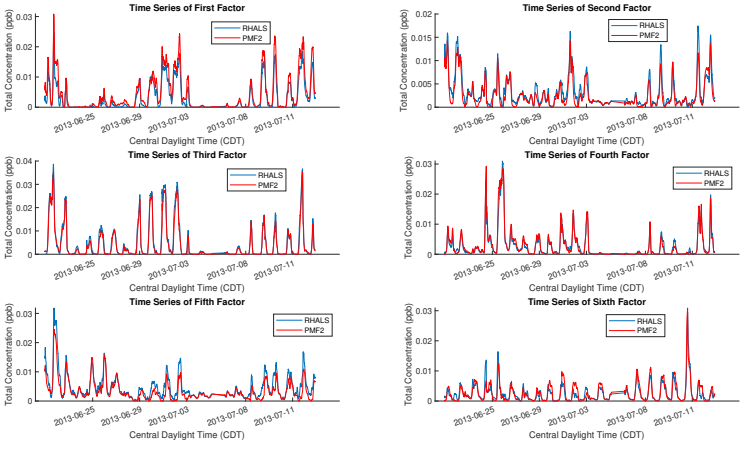

Figure 13. Two hour rolling average of RHALS time series factors overlaid with PMF2 factors.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

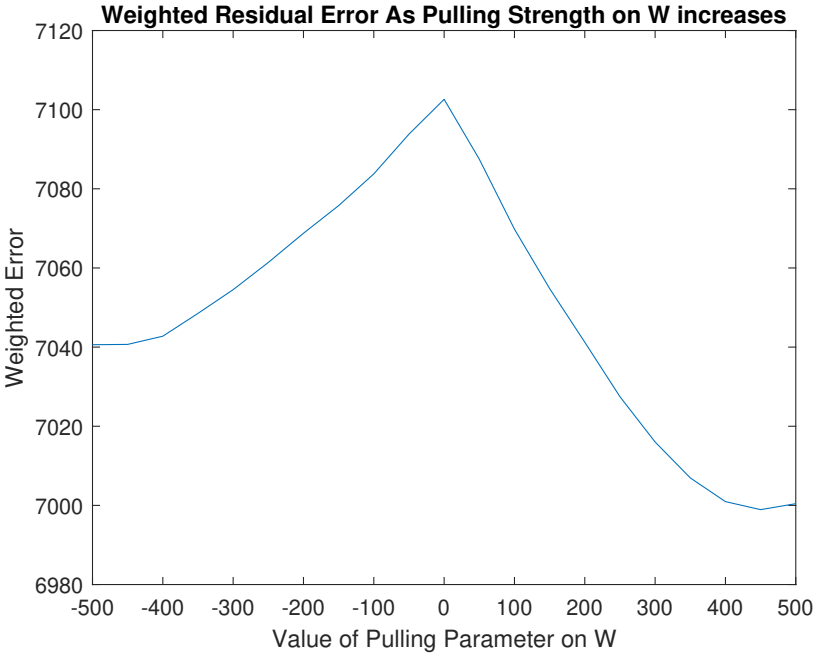

Figure 14. Weighted residual error of rotated solutions from the 13^{th} trial of RHALS.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

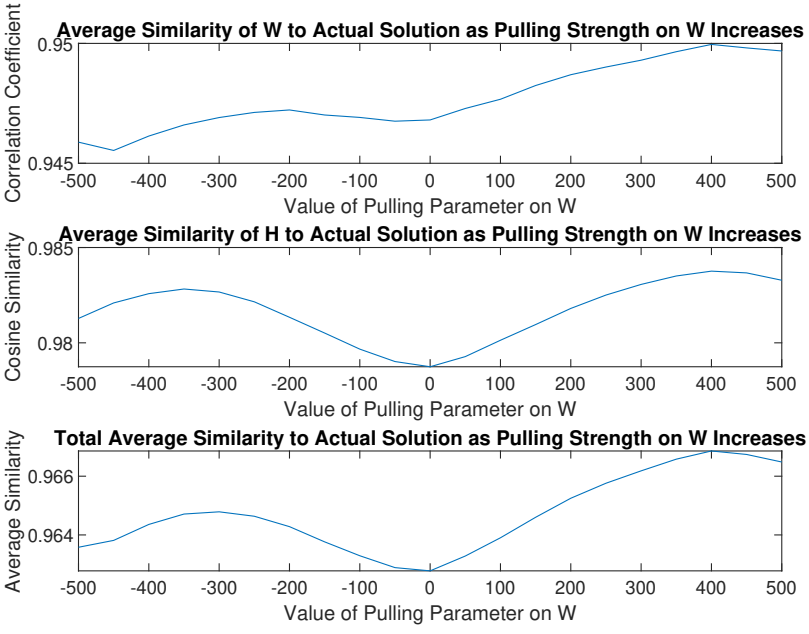

Figure 15. Similarities to PMF2 solution of rotated solutions from the 13^{th} trial of RHALS.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

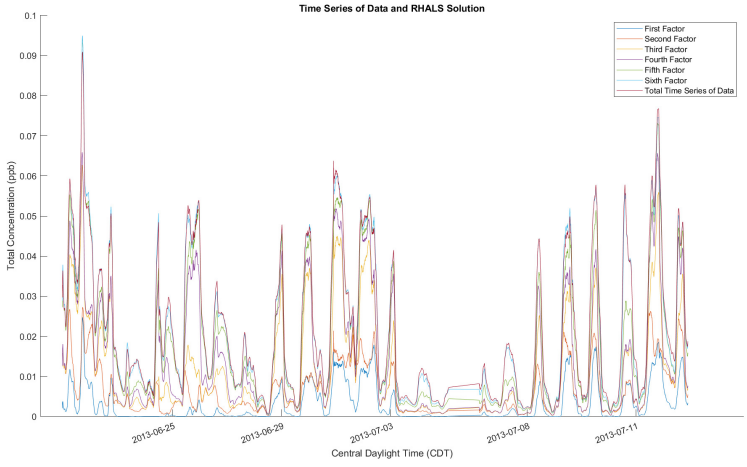

Figure A1. RHALS Time Series Factors laid out with total time series from data, plotted as a two hour rolling average.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

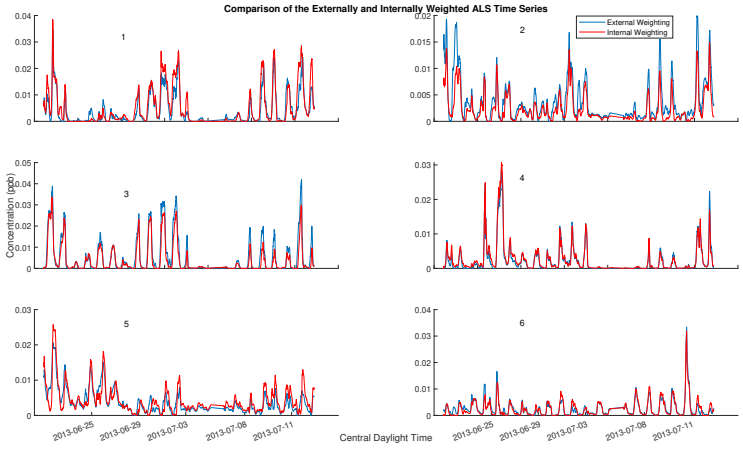

Figure A2. Comparison of two hour rolling average of time series for externally Weighted and internally weighted ALS factors. Externally weighted error: 7.2465×10^3. Internally weighted error: 6.4853×10^3.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

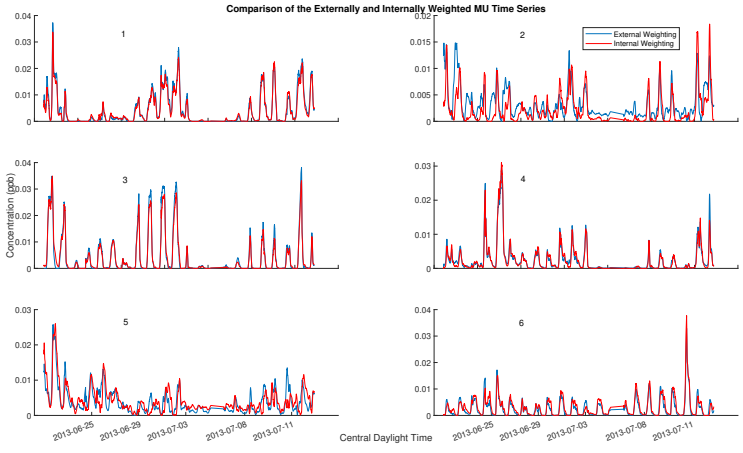

Figure A3. Comparison of two hour rolling average of time series for externally Weighted and internally weighted MU factors. Externally weighted error: 7.0239×10^3. Internally weighted error: 6.3900×10^3.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

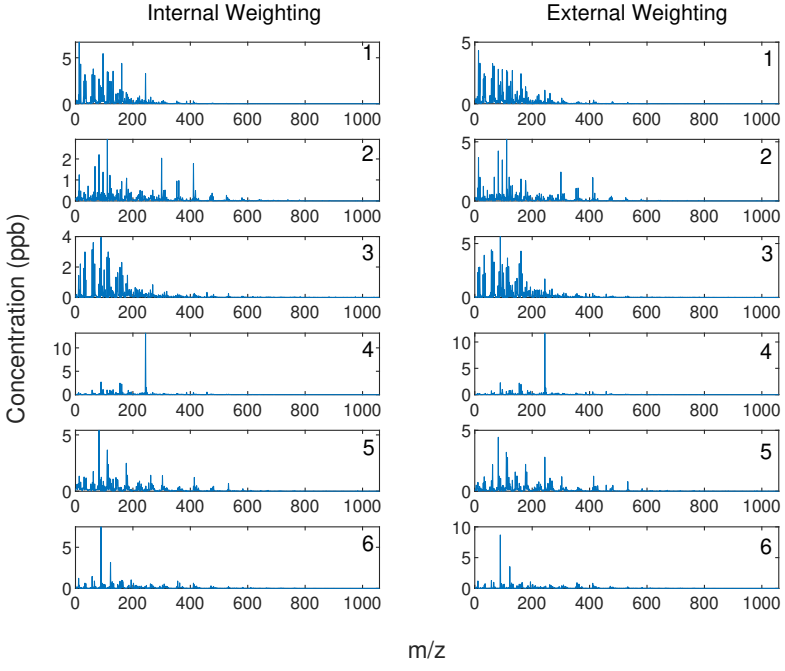

Figure A4. Comparison of mass spectra for externally weighted and internally weighted ALS factors.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

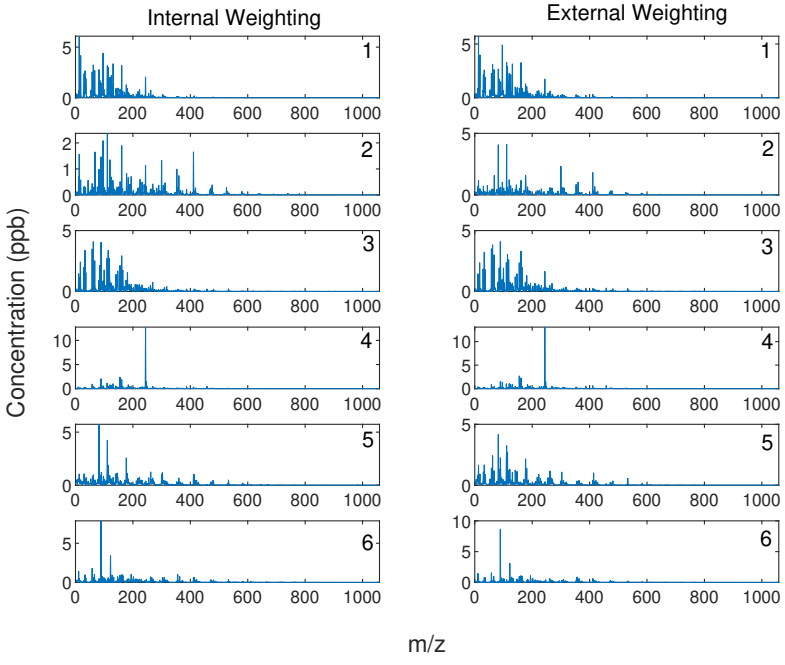

Figure A5. Comparison of mass spectra for externally weighted and internally weighted MU factors.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

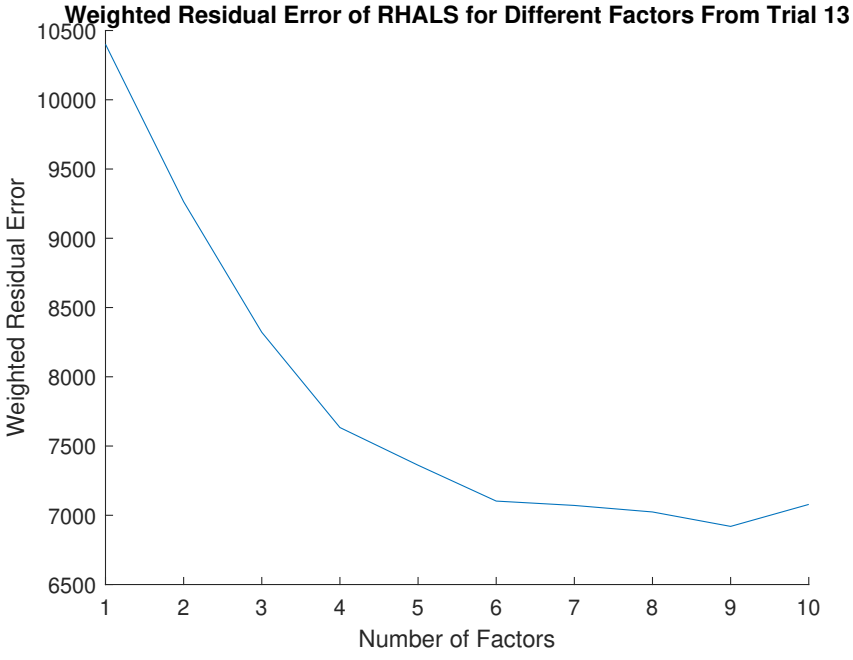

Figure A6. Number of factors versus weighted residual error for the RHALS algorithm.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

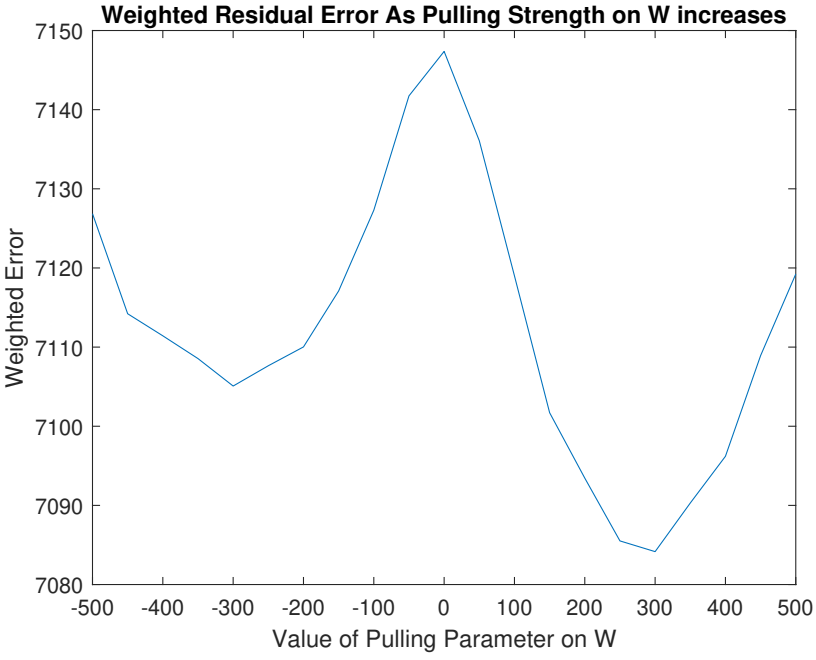

Figure A7. Weighted residual error of rotated solutions from the 11^{th} trial of RHALS.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

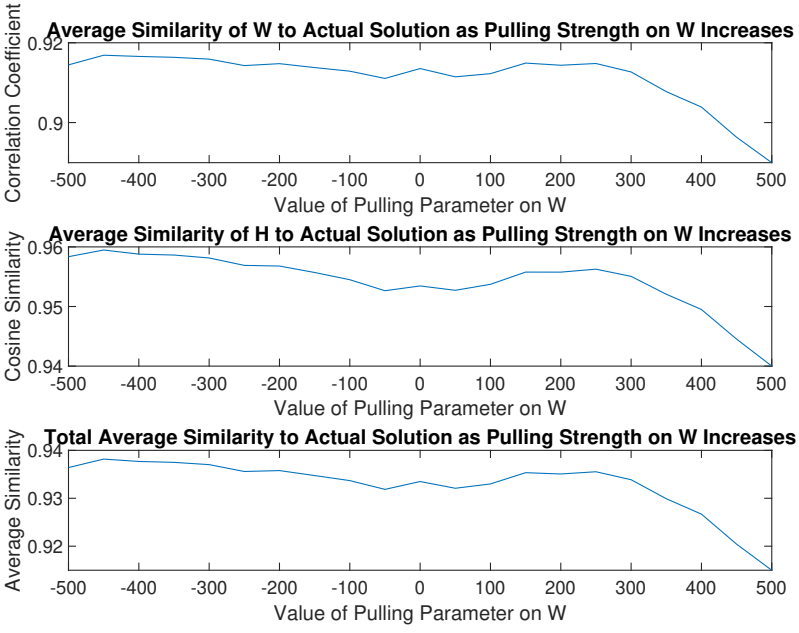

Figure A8. Similarities to the PMF2 solution of rotated solutions from the 11^{th} trial of RHALS. The total similarity refers to an average of the similarity metrics for the time series and mass spectra factors.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

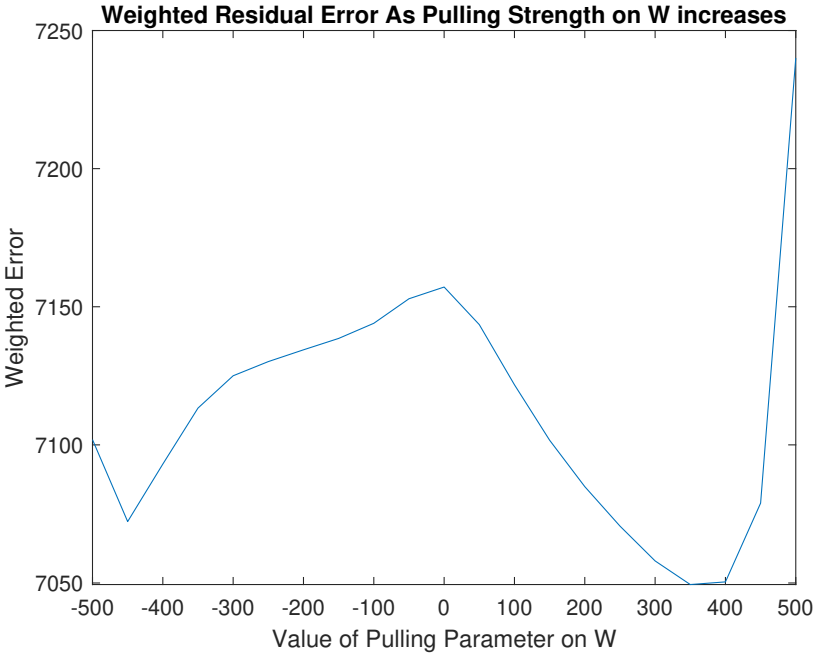

Figure A9. Weighted residual error of rotated solutions from the 17^{th} trial of RHALS.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

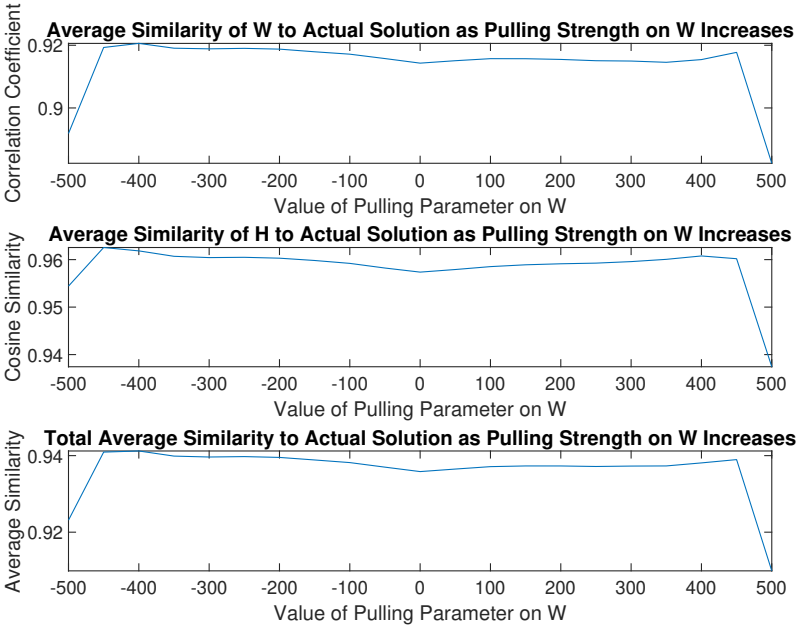

Figure A10. Similarities to PMF2 solution of rotated solutions from the 17^{th} trial of RHALS.

https://doi.org/10.5194/egusphere-2022-1221
Preprint. Discussion started: 20 December 2022

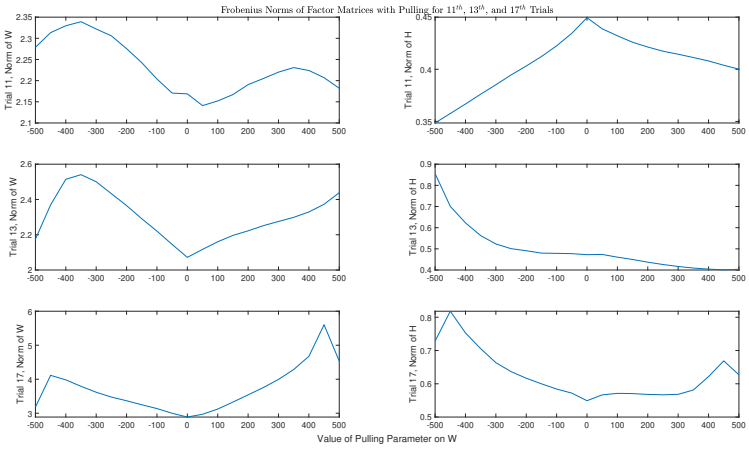

Figure A11. Frobenius norms of rotated factor matrices for the 11^{th}, 13^{th}, and 17^{th} trials.