2006-05-02

# Bayesian and Positive Matrix Factorization approaches to pollution source apportionment

Jeff William Lingwall
*Brigham Young University - Provo*

# BAYESIAN AND POSITIVE MATRIX FACTORIZATION APPROACHES TO POLLUTION SOURCE APPORTIONMENT

by

Jeff W. Lingwall

A Thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Statistics

Brigham Young University

August 2006

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a Thesis submitted by

Jeff W. Lingwall

This Thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date — Dr. William F. Christensen, Chair

Date — Dr. Shane Reese

Date — Dr. David Whiting

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the Thesis of Jeff W. Lingwall in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Dr. William F. Christensen
Chair, Graduate Committee

Accepted for the Department

Scott Grimshaw
Graduate Coordinator

Accepted for the College

Tom Sederberg
Associate Dean, College of Physical and Mathematical Sciences

# ABSTRACT

# BAYESIAN AND POSITIVE MATRIX FACTORIZATION APPROACHES TO POLLUTION SOURCE APPORTIONMENT

Jeff W. Lingwall

Department of Statistics

Master of Science

The use of Positive Matrix Factorization (PMF) in pollution source apportionment (PSA) is examined and illustrated. A study of its settings is conducted in order to optimize them in the context of PSA. The use of *a priori* information in PMF is examined, in the form of target factor profiles and pulling profile elements to zero. A Bayesian model using lognormal prior distributions for source profiles and source contributions is fit and examined.

# Acknowledgements

The best way to become acquainted with a subject is to write a book about it. —Benjamin Disraeli

This thesis would not have been possible without help from many people. First, my wife Julia, for not minding long hours working and studying. Dr. Christensen, for letting me bother him for two years and being a true mentor. My graduate committee, especially Dr. Reese, for their help and advice. Dr. Jay Turner and Dr. James Schauer for access to data from the St. Louis Supersite. And finally, all the faculty and students of the Department of Statistics who have either helped me, taught me, or influenced me.

# Contents

**Chapter**

**Appendix**

# Tables

Table

## Figures

Figure

# Chapter 1

## Introduction

Pollution source apportionment (PSA) is the practice of deriving information about pollution sources and the amount they emit from ambient air pollution data. In an industrial society that pollutes its air, a reliable and accurate pollution source apportionment model would allow regulating agencies to know which sources are contributing to the airshed and in what amounts. Since air pollution has been linked to mortality (Dockery *et al.*, 1993), knowledge of what is contributing to the problem is important. In an article on the contribution of statistics to environmental epidemiology, Thomas (2000) discusses PSA and notes that, for epidemiology in general, "greater attention by statisticians to these problems in particular would certainly help advance the field."

Many methods for approaching the PSA problem exist, based on various statistical techniques and differing amounts of information that can be assumed about the number of polluting sources and their compositions. The basic model followed is

$$\underset{p\times n}{\mathbf{Y}} = \underset{p\times k}{\mathbf{\Lambda}}\underset{k\times n}{\mathbf{F}} + \underset{p\times n}{\boldsymbol{\epsilon}} \tag{1.1}$$

where $\mathbf{\Lambda}$ is a matrix containing pollution source profiles for the $k$ sources and $p$ chemical species, $\mathbf{F}$ is a matrix of the $k$ sources' contribution to the airshed over $n$ time periods, and $\mathbf{Y}$ is a matrix of measurements on $p$ different chemical species observed at $n$ times. For example, the concentration of species $i$ observed at time $j$, $y_{ij}$, measured at a receptor can be explained as

$$y_{ij} = \boldsymbol{\lambda}_i \mathbf{f}_j \tag{1.2}$$

where $\mathbf{f}_j$ is the $j$th column of $\mathbf{F}$ and $\boldsymbol{\lambda}_i$ is the $i$th row of $\mathbf{\Lambda}$ (see Christensen and Sain, 2002).

The problem is difficult, for many reasons. For instance, the number of pollution sources is really uncountable, since every car on the road emits chemicals in differing amounts, every house with a fireplace burns wood on some days and not on others, etc. When the pollution sources are assumed to be known, they may change over time as factories change output, close, or as emission standards for vehicles are tightened. Ambient air measurements can be contaminated as wind brings in pollutants from other areas, making the very thought of an "airshed" difficult. An additional constraint in all solutions is that all elements must be positive, since negative source profiles or source contributions are not realistic.

Despite these problems, assumptions can be made that lead PSA to be tractable. For example, even though we do not know the exact source profile of every car on the road, we can assume that cars generally emit the same type of profile and that emissions might be higher on weekdays instead of weekends. Tracer sources can be identified that can validate the mathematical model. The number of pollution sources considered can be limited to “major” sources for reasonable interpretation. These assumptions can lead to results that appear to fit reasonable models for airsheds.

As mentioned above, various statistical methods exist for approaching the PSA problem. Differing methods have strengths and weaknesses, although some are more widely used and studied than others. Figure 1.1 shows a range of different techniques on a scale of little-information to perfect-information. The two methods examined in this thesis are Positive Matrix Factorization (PMF) and Bayesian methods.

Positive Matrix Factorization is a variant of factor analysis that is becoming widely used in PSA. In contrast to traditional factor analysis methods which decompose $\mathbf{Y}$ based on the correlation matrix, PMF solves the factor analysis equations by iteratively computing $\mathbf{F}$ and $\boldsymbol{\Lambda}$ via the minimization of the (simplified) equation

$$Q = \sum_{i=1}^{n} \sum_{j=1}^{p} e_{ij}^2 / s_{ij}^2 \tag{1.3}$$

where $e_{ij}$ is calculated as $y_{ij} - \boldsymbol{\lambda}_j \mathbf{f}_i$ and $s_{ij}$ is the standard deviation associated with each data point (Paatero and Tapper, 1994; Paatero, 1997). PMF constrains the results to be positive, thus satisfying the non-negativity constraints necessary for realistic pollution source apportionment models. PMF is essentially a weighted least squares solution to the equations (Paatero, 1997).

In a Bayesian approach to PSA, the problem can be approached in various ways. If one assumes knowledge of the pollution source profiles, $\boldsymbol{\Lambda}$, then a Bayesian version of the Chemical Mass Balance model can be fit, using prior distributions on

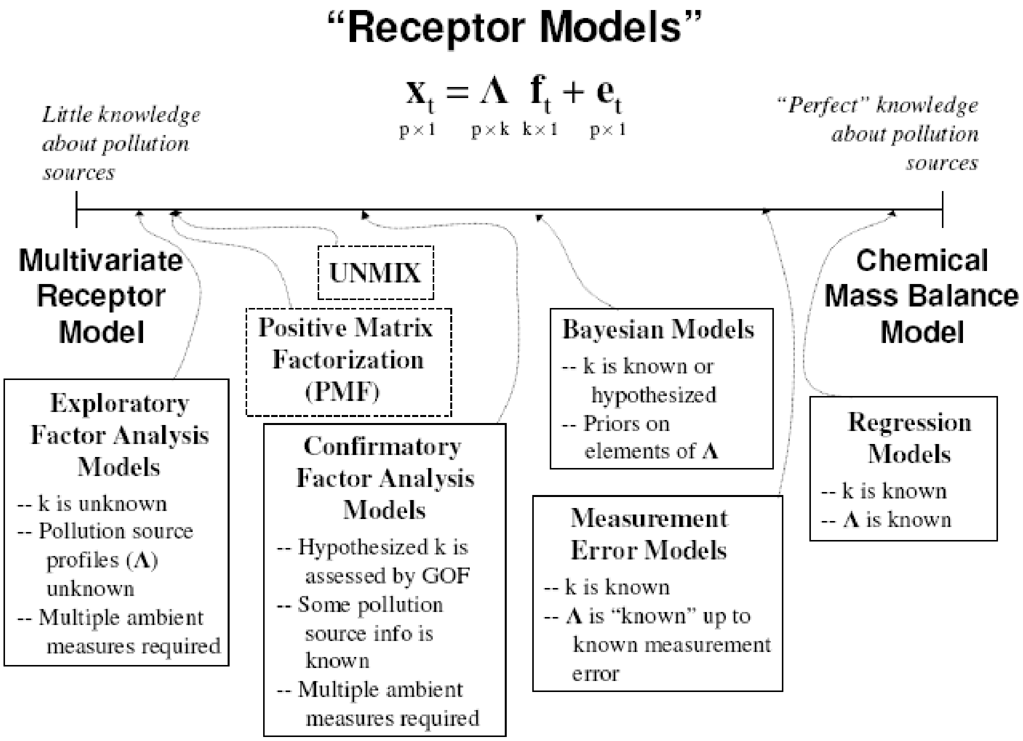

"Receptor Models"
$\underset{p\times 1}{\mathbf{x}_t} = \underset{p\times k}{\Lambda}\ \underset{k\times 1}{\mathbf{f}_t} + \underset{p\times 1}{\mathbf{e}_t}$
Little knowledge about pollution sources
"Perfect" knowledge about pollution sources
Multivariate Receptor Model
Chemical Mass Balance Model
UNMIX
Positive Matrix Factorization (PMF)
Exploratory Factor Analysis Models
-- k is unknown
-- Pollution source profiles (Λ) unknown
-- Multiple ambient measures required
Confirmatory Factor Analysis Models
-- Hypothesized k is assessed by GOF
-- Some pollution source info is known
-- Multiple ambient measures required
Bayesian Models
-- k is known or hypothesized
-- Priors on elements of Λ
Measurement Error Models
-- k is known
-- Λ is "known" up to known measurement error
Regression Models
-- k is known
-- Λ is known

Figure 1.1: Some methods used in pollution source apportionment (Christensen *et al.*, 2006)

elements of $\mathbf{F}$, with parameters estimated using Markov chain Monte Carlo methods. If one assumes less knowledge of the pollution source profiles, prior distributions can be fit to elements of $\boldsymbol{\Lambda}$ and both $\mathbf{F}$ and $\boldsymbol{\Lambda}$ may be estimated. The approach taken in this work lies between these two and assumes some knowledge of $\boldsymbol{\Lambda}$ while still letting the algorithm estimate $\hat{\boldsymbol{\Lambda}}$.

# Chapter 2

# Review of Literature

## 2.1 Pollution source apportionment

The use of statistical methods in pollution source apportionment dates back to at least 1967, when factor analysis was applied to pollution data from 30 cities in the United States measured from 1957 to 1961 on 13 different chemical species. Blifford and Meeker (1967) identified four different factors using factor analysis, including Varimax, Quartimax, and Oblimax rotations. The factors they identified were heavy industry, automobiles, fuel burning, and petroleum refining. They regressed their factor scores on the 30 cities as a check, and compared the results to known knowledge about the cities (Blifford and Meeker, 1967).

This multivariate approach was continued, including a paper nine years later by Hopke *et al.* (1976), where data collected over five months from Boston were analyzed using factor analysis and cluster analysis. The researchers made improvements over Blifford and Meekers' (1967) method and identified a key problem of the factor analysis approach: "It will often fall to the judgment of the investigators as to how many factors to keep." They also succinctly stated an advantage to the use of factor analysis, that it “requires no a priori assumptions about the compositions or even the total number of components” (Hopke *et al.*, 1976).

An approach different from the multivariate approach appeared in Miller *et al.* (1972). This approach assumed that the “mass of material from source $j$ per unit mass of aerosol”, summed over $j$ sources must be equal to one. Using tracer elements identified in each source, the researchers were able to solve equations to estimate the source contributions. This “chemical element balance” method developed into the widely used CMB approach.

Many different approaches have been developed over the years to tackle this problem. One important improvement that new methods have tried to develop is to reduce the indeterminacy of the factor analysis model, so that less arbitrary models could be extracted from the data. Some examples of these methods are UNMIX (Henry, 2003), ICFA (Christensen *et al.*, 2006), iterative TTFA (Henry, 1991), and PMF (Paatero and Tapper, 1994).

UNMIX, a geometrical approach, finds hyperplanes in $n$-dimensional space to fit a receptor model (Henry, 2003). Iterative TTFA “transforms . . . eigenvectors to minimize the difference between the transformed vector and a target vector.” An iterative process is used so that knowledge of the source profiles need not be precise (Henry, 1991).

Iterated Confirmatory Factor Analysis (ICFA) is a variant of factor analysis that produces a unique solution via a method similar to the EM algorithm. After initial estimates are obtained, some elements of the source profile matrix are treated as fixed, while others are estimated holding the fixed elements constant. This process is iterated until convergence is reached (Christensen *et al.*, 2006).

## 2.2 Positive Matrix Factorization

In 1994, Paatero and Tapper introduced PMF (Paatero and Tapper). As stated before, they diverged from traditional factor analysis by not using the correlation matrix in their algorithm. Instead, they find $\mathbf{\Lambda}$ and $\mathbf{F}$ through a process of minimization, as shown in Equation (3.3). They chose to use the name "Positive Matrix Factorization" to avoid the term "Factor Analysis," which is used in different ways by statisticians and non-statisticians, and "Principle Component Analysis," since they do not use singular value decomposition to solve the equations.

In PMF, different algorithms can be used to solve the equations. For example, an early algorithm used a modification of Alternating Regression (AR). In AR, in our case, $\mathbf{F}$ would be held constant while $\mathbf{\Lambda}$ is estimated, and then $\mathbf{\Lambda}$ would be held constant while $\mathbf{F}$ is estimated. Paatero and Tapper improved on this method by introducing a third step, one that estimated $\mathbf{F}$ and $\mathbf{\Lambda}$ simultaneously. The non-negativity constraints were handled by setting negative elements to zero and maximizing a penalty function associated with each data point (Paatero and Tapper, 1994).

PMF offers several advantages over PCA and exploratory factor analysis. For example, in traditional exploratory factor analysis the solutions are guaranteed to be non-unique, since any orthogonal matrix $\mathbf{T}$ can be introduced into the equations without changing the results, causing a “rotation”, as shown in Equation (2.1).

$$\mathbf{Y} = \mathbf{\Lambda} \mathbf{T}' \mathbf{T} \mathbf{F} + \boldsymbol{\epsilon} \tag{2.1}$$

In contrast, the results of PMF may be unique, when a “significant” number of zeros appear in all the columns of **F** and **Λ** (Paatero and Tapper, 1994). Other advantages include the handling of outliers and the ability to handle missing values.

Three years later, Paatero (1997) introduced PMF2, a revised version of PMF (PMF2 handles two dimensional arrays, while PMF3 handles three dimensional arrays). PMF2 converges faster than PMF, and when studied will be referred to simply as PMF.

There is a large literature about the application of PMF, in both PSA and other areas. For example, Hopke *et al.* (1999) analyzed particulate data from northern Canada using PMF and another PMF program by Paatero, the Multilinear Engine (ME) . Lee *et al.* (1999) analyzed data gathered in Hong Kong from 1992 to 1994. They used several outputs from PMF ($Q$, the scaled residual matrix, and $rotmat$) to select the number of factors. Recently, Paatero *et al.* (2003) incorporated temperature, wind velocity vectors, and other environmental information into an analysis of pollution in the eastern United States using the ME.

## 2.3 Bayesian methods

Bayesian statistics began with a paper by Thomas Bayes (1763), *An essay towards solving a problem in the doctrine of chances*. The problem he undertook to solve was stated as

> *Given* the number of times in which an unknown event has happened and failed: *Required* the chance that the probability of its happening in a single trial lies somewhere between any two degrees of probability that can be named.

Bayes saw that one could solve the problem by

> estimat[ing] the chance that the probability for the happening of an event . . . should lie between any two named degrees of probability, antecedently to any experiments made about it.

By incorporating prior information into the problem, solutions could be found (Dale, 1999).

The practical application of Bayesian statistics was limited for many years due to the intractability of the mathematics. Calculating normalizing integrals in high-dimensional space for practical applications simply is not practical. This limited the practical application of Bayesian statistics until the advent of numerical methods and the computer. Solutions that were impossible to find analytically could be approximated to high degree via simulation. With cheap computing and the power of numerical methods, Bayesian statistics are now widely used and accepted.

Various simulation methods exist, from rejection sampling to the Metropolis algorithm. Markov chain Monte Carlo algorithms use Monte Carlo integration, where samples are used to approximate a distribution, and Markov chains, drawing the samples from a Markov chain with each state dependent on the previous (Gilks *et al.*, 1996). The Metropolis algorithm, used in this thesis, is one example.

The basic Metropolis algorithm seeks a Markov chain that has a stationary distribution equal to the one we seek. The Metropolis algorithm is a special case of the Metropolis-Hastings algorithm that uses only symmetric jumping distributions, which also yields easy interpretation, "the chain moves to higher $\pi$-density regions automatically but only with appropriate probabilities to lower regions" (Billera and Diaconis, 2001).

Bayesian methods have been applied to subjects as diverse as species loss in the Amazon (Ferraz *et al.*, 2003), gene expression change in honey bees (Grozinger *et al.*, 2003), hospital quality (Geweke *et al.*, 2003), three dimensional image reconstruction (Eppstein *et al.*, 2002), and phylogenetic analysis (Miller *et al.*, 2002). Bayesian methods apply themselves easily to complex, multi-dimensional problems and often provide much more useful information than a frequentist approach.

Bayesian methods have began to be applied to pollution problems. In 2000, researchers from Johns Hopkins University fit a hierarchical model to PM10 data from 20 cities and found a relationship with mortality. They formed time series of the mortality and pollutants in the cities and then used regression to find pollution rates. Next, they regressed these rates on city variables and used Markov chain Monte Carlo methods to find pollution effects. They found a "consistent association" between the PM10 rates and daily mortality, and, while they don't claim to have established causality, they hint at it (Dominici *et al.*, 2000).

Park *et al.* (2001) published a Bayesian analysis of data from Atlanta, emphasizing the temporal dependence of the data. Their approach is to assume the number of sources and certain identifiability conditions in order to fit a model. Using the judgment of an environmental engineer and source measurement, they established zeros in the source profiles and use MCMC to fit a hierarchical model. They assumed normally distributed errors in their analysis and leave the study of heavy-tailed errors open. Our current study using lognormal distributions, which have heavy-tails, may help open this area to study.

# Chapter 3

# The optimization of PMF

## 3.1 Pollution source apportionment

Pollution source apportionment is the practice of deriving information about pollution sources and the amount they emit from ambient air pollution data. Various methods are employed, based on differing amounts of information that can be assumed about the number of polluting sources and their compositions. Factor analysis techniques can be used when the pollution sources are unknown, using the equation

$$\mathbf{Y} = \boldsymbol{\Lambda}\mathbf{F} + \boldsymbol{\epsilon} \tag{3.1}$$

where $\boldsymbol{\Lambda}$ is a $k \times p$ matrix containing pollution source profiles for the $k$ sources, $\mathbf{F}$ is a $k \times n$ matrix of the sources' contributions to the airshed, and $\mathbf{Y}$ is an $p \times n$ matrix of measurements of $p$ different chemical species observed at $n$ times. Thus, for example, the concentration of species $i$ observed at time $j$, $y_{ij}$, measured at a receptor can be explained as

$$\underset{1\times 1}{y_{ij}} = \underset{1\times k}{\boldsymbol{\lambda}_i}\,\underset{k\times 1}{\mathbf{f}_j} \tag{3.2}$$

where $\mathbf{f}_j$ is the $j$th column of $\mathbf{F}$ and $\boldsymbol{\lambda}_i$ is the $i$th row of $\boldsymbol{\Lambda}$ (see Christensen and Sain, 2002).

## 3.2 Positive Matrix Factorization

Traditional factor analysis, however, fails in an important aspect of the pollution source apportionment problem. In traditional factor analysis, there are no non-negativity constraints on the results. Since in air pollution studies realistic results must be non-negative, traditional factor analysis fails to provide the optimal solution. Alternatives exist, one of which is Positive Matrix Factorization.

PMF (Paatero and Tapper, 1994) is a method related to factor analysis. In contrast to traditional factor analysis methods which decompose $\mathbf{Y}$ based on the correlation matrix, PMF solves the factor analysis equations by iteratively computing $\mathbf{F}$ and $\boldsymbol{\Lambda}$ via the minimization of

$$Q = \sum_{i=1}^{n} \sum_{j=1}^{p} e_{ij}^2 / s_{ij}^2 \tag{3.3}$$

where $e_{ij}$ is calculated as $y_{ij} - \boldsymbol{\lambda}_j \mathbf{f}_i$ and $s_{ij}$ is the standard deviation associated with each data point (Paatero and Tapper, 1994). PMF constrains the results to be positive, thus satisfying the non-negativity constraints necessary for realistic pollution source apportionment models. In 1997, Paatero introduced a revised PMF algorithm called PMF2, which is examined in this paper and will be referred to as PMF (Paatero, 1997).

PMF can be daunting for the novice user. Many different settings are available to influence the output of the program. This section examines some of these settings, attempts to optimize each of them, and makes some suggestions for the beginning user. The PMF settings considered here include:

- The use of **T** and **V** in the computation of the array of standard deviations denoted by **S**
- The use of EM=-10,-12,-13, and -14 in the computation of **S**
- The use of C1 and C3 in the computation of **S**
- Fpeak
- Outlier Threshold Distance

### 3.3 Research method

Six hundred simulated data sets were created, based on five sources: sea salt, secondary sulfate, winter secondary, soil, and auto/diesel. Each simulated observation constituted measurements on 23 different chemical species. The five source profiles were combined to create an array of source profiles, $\mathbf{\Lambda}$.

Instead of randomly creating the source contributions, **F**, data that was reflective of the real world was used based on an analysis of an actual data set from Washington, D.C.. The results from the analysis were treated as if they were actually the source contributions. Thus, a natural degree of temporal correlation, seasonality, and daily and weekly fluctuations are built into the data. Although these five source contributions represent only one possibility of the wildly varying airsheds that exist in actual analysis, they allow a starting point that reflects real sources for study. These source contributions and profiles are shown in Figures 3.1 and 3.2, respectively.

Two different types of simulated data were considered. First, based only on the five sources listed above, and secondly, with the addition of three minor sources. These three minor sources, a wood source, lead smelter source, and an additional secondary source, would serve as unidentified minor sources in the model, or as contamination. These three minor sources are shown in Table 3.1.

Then, for the uncontaminated simulation, data with three different degrees of noise was created, where an observation $y_{ij}$ was obtained as a draw from a lognormal distribution with mean $\boldsymbol{\lambda}_i \mathbf{f}_j$ and a coefficient of variation (CV) of either 0.2, 0.5, or 1.0.

For the contaminated simulation, each observation $y_{ij}$ was obtained as a draw from a lognormal distribution with mean $\boldsymbol{\lambda}_i^* {\mathbf{f}_j}^*$, where $\boldsymbol{\lambda}_i^*$ is the $i$th row of $\boldsymbol{\Lambda}^*$ which contains the three minor source profiles in addition to the five major sources, and $\mathbf{f}_j^*$ is the $j$th column from $\mathbf{F}^*$, which contains the three minor source contributions as well as the five major sources. The three minor sources were down-weighted by multipliers of 0.2 or 0.5. As with the uncontaminated simulation, three different CV's were used: 0.2, 0.5, and 1.0.

One hundred data sets of each combination of CV and contamination were created, as illustrated in Table 3.2, resulting in 600 simulated data sets representing a wide range of potential airsheds.

| **Element** | **Wood** | **Secondary** | **Lead smelter** |
|---:|---:|---:|---:|
| Al | 0.00113 | 0.0087 | 0.00174 |
| As | 0 | 0.00273 | 0.02905 |
| Br | 0 | 0.00063 | 0.00157 |
| Ca | 0.00113 | 0.02161 | 0.01337 |
| Cl | 0.0133 | 0.03419 | 0.0164 |
| Cu | 0 | 0.00467 | 0.05425 |
| EC | 0.39645 | 0.18306 | 0.55505 |
| Fe | 0 | 0.00907 | 0.02222 |
| K | 0.03389 | 0.00269 | 0.00343 |
| Mn | 0 | 0.00033 | 0.00026 |
| Na | 0.00452 | 0.07802 | 0 |
| Ni | 0 | 0.01011 | 0.00156 |
| NO3 | 0.02259 | 0.18306 | 0 |
| OC | 0.50837 | 0.18306 | 0 |
| P | 0 | 0.00059 | 0.00013 |
| Pb | 0 | 0.02338 | 0.24938 |
| Se | 0 | 0.00011 | 0.00026 |
| Si | 0.00339 | 0.05063 | 0.01908 |
| SO4 | 0.01356 | 0.18306 | 0 |
| Sr | 0 | 0 | 0.00053 |
| Ti | 0 | 0.00156 | 0.00021 |
| V | 0 | 0.01216 | 0 |
| Zn | 0.00167 | 0.0066 | 0.0315 |

Table 3.1: Minor sources introduced as contamination and not identified in the model.

| **Contamination** | **CV** | | |
|:---:|:---:|:---:|:---:|
| uncontaminated | 0.2 | 0.5 | 1.0 |
| contaminated | 0.2 | 0.5 | 1.0 |

Table 3.2: Combinations of CV and contamination used to create simulated data

After the creation of the simulated data, PMF was used to estimate $\boldsymbol{\Lambda}$ and $\mathbf{F}$ for each data set. Since PMF's output does not necessarily match the researcher's ordering of the profiles and contributions, the resulting rows of $\hat{\mathbf{F}}$ and columns of $\hat{\boldsymbol{\Lambda}}$ were then sorted. Factor elements were sorted based on a minimization of the MSE between rows of $\hat{\mathbf{F}}$ and rows of $\mathbf{F}$.

After PMF's results had been sorted, the average absolute error (AAE) was calculated between PMF's results and the original matrices. AAE is a metric used in source apportionment, using the absolute value of the error instead of the squared error (see Javitz, Robinson, and Watson, 1988, and Christensen and Gunst, 2004). The AAE for $\hat{\boldsymbol{\Lambda}}$ was calculated as

$$AAE = \frac{1}{p} \sum_{i=1}^{p} \sum_{h=1}^{k} |\hat{\lambda}_{ih} - \lambda_{ih}| \tag{3.4}$$

where $p = 23$ is the number of chemical species in the source profile, and $k = 5$ is the number of profiles in $\boldsymbol{\Lambda}$. AAE is thus simply taking an average of the absolute error $|\hat{\lambda}_{ih} - \lambda_{ih}|$ over $i$ for each of the $k$ sources and summing them over $k$. The AAE of $\hat{\mathbf{F}}$ was calculated in a similar manner for the source contributions.

### 3.4 Default settings

Under the default settings, PMF succeeds in converging with the simulated data. One small change was made, to scale the sum of the source profile columns to one. This assumes that we have all the essential chemical species in the model captured, so that we are accounting for 100% of the pollution source. Additionally,

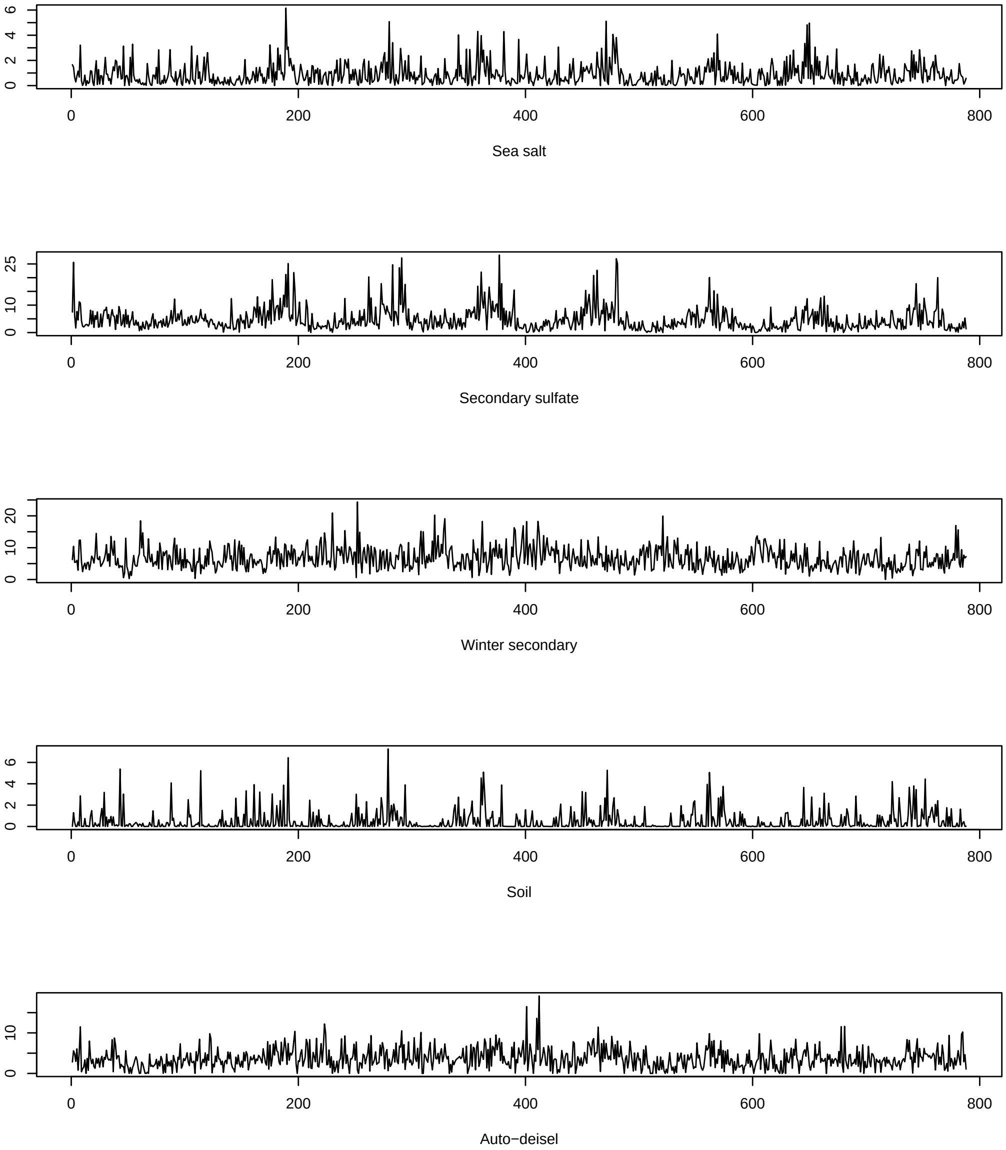


Figure 3.1: True source contributions for simulated data (**F**)

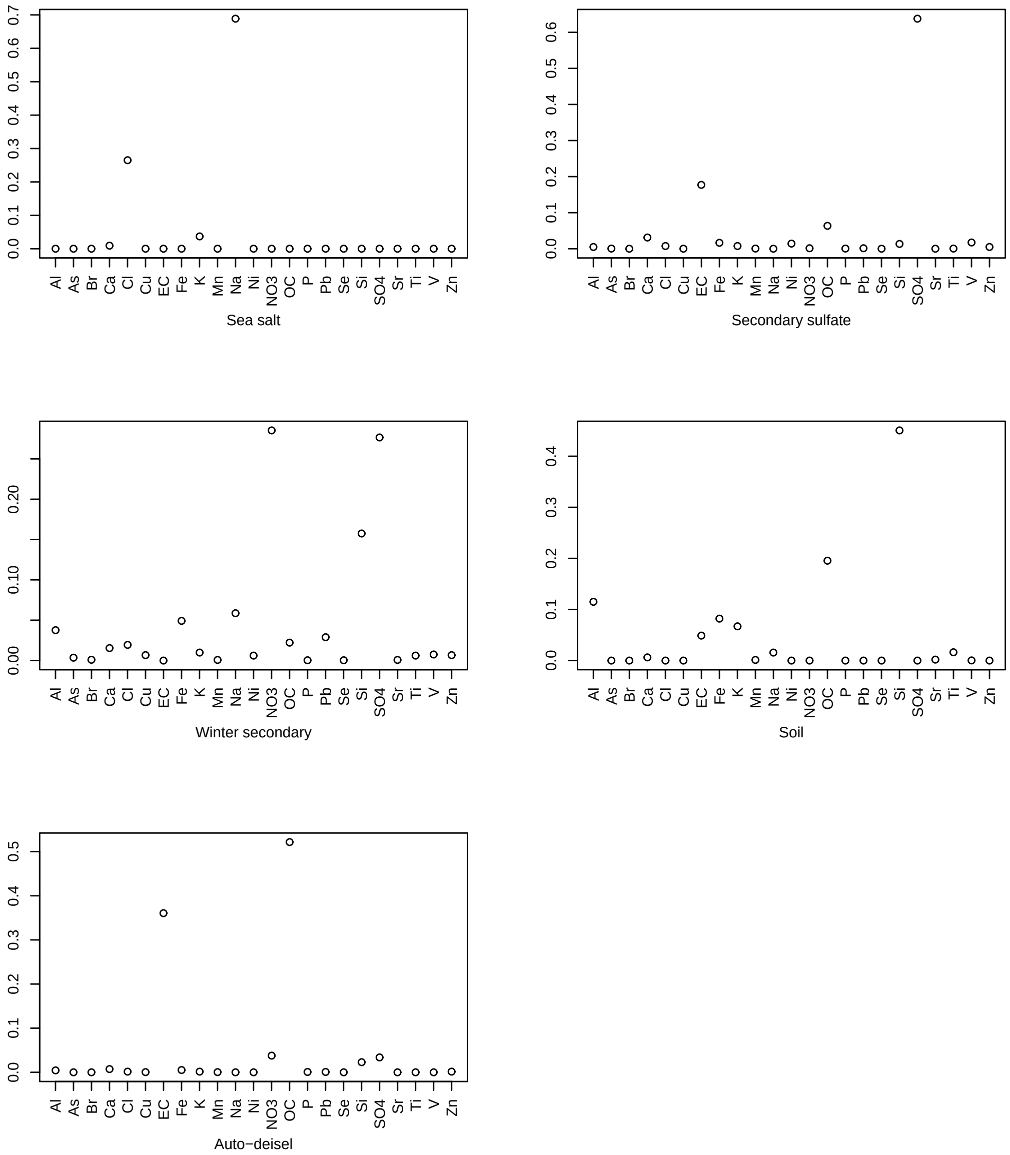


Figure 3.2: True source profiles for simulated data ($\mathbf{\Lambda}$)

this allows for better comparison with the true source profiles, which have columns that sum to one.

Figures 3.3 and 3.4 show the results over the default settings. As can be seen, PMF performs almost as one would expect. The AAE for $\hat{\mathbf{F}}$ increases as we increase the CV, which we expect because as we increase the CV we introduce more noise into the data. The AAE for $\hat{\boldsymbol{\Lambda}}$ similarly increases as we increase the CV. In the CV = 0.2 and 0.5 results, the contaminated data, where we have introduced minor sources not included in the model, is clearly differentiated from the uncontaminated data, much more for the estimates of the source profiles, $\boldsymbol{\Lambda}$, than the source contributions, $\mathbf{F}$. This indicates that if we do not account for all of the sources in our model, our estimates of the profiles will be more erroneous than our estimates of their contributions.

Interestingly, when we increase the CV to 1.0, there is little difference between the results for the contaminated data and the results for the uncontaminated data. It seems that when the data is quite noisy, it matters less to identify all the sources, since the high noise blurs out precise identifiability, or that with high noise the influence of minor sources simply disappears.

## 3.5 Optimizing settings

As mentioned earlier, PMF offers many settings that may be changed to influence the performance of the algorithm. This section addresses whether certain settings are better as applied to our ambient particulate data. In practice, it seems that these settings are used in a haphazard manner, so this simulation study sheds

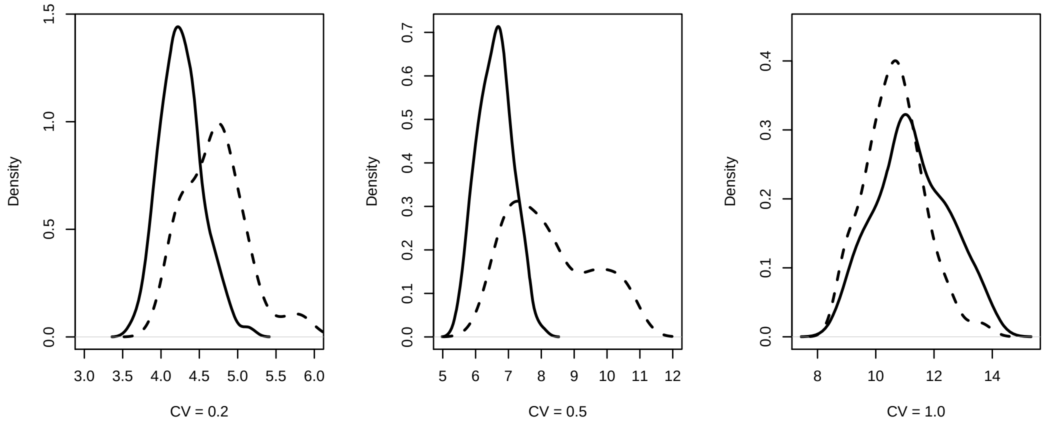


Figure 3.3: Kernel density estimates of AAE for $\hat{\mathbf{F}}$ over PMF's default settings. Contaminated data shown in dashed lines.

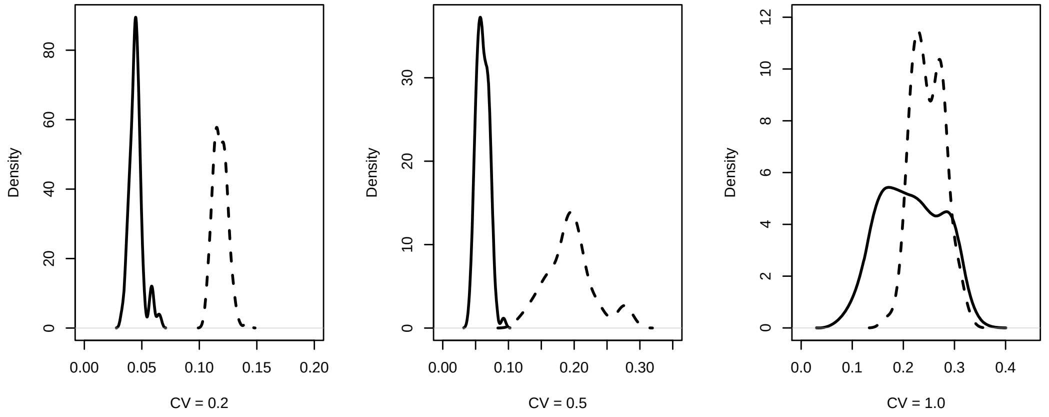


Figure 3.4: Kernel density estimates of AAE for $\hat{\mathbf{\Lambda}}$ over PMF's default settings. Contaminated data shown in dashed lines.

light on whether the settings actually change results, and, if so, how the settings might be optimized in the context of pollution source apportionment.

### 3.5.1 Outlier threshold distance

PMF uses the *outlier threshold distance* to control what it processes as an outlier. The default value provided by the program is 4. The PMF manual gives this information about the outlier threshold distance (Paatero, 2004b)

> The "processing as an outlier" means that the std-dev value $s_{ij}$ or $s_{ijk}$ is increased so that the "pull" or influence of the outlying value $x_{ij}$ or $x_{ijk}$ is no more than the pull of a value which is on the limit of being classified as an outlier. This corresponds to the Huber estimation principle. —We suggest that the following values should be used as outlier threshold distance: $\alpha$= 2.0, 4.0, or 8.0. By adhering to these standard values one makes it easier to compare PMF results obtained by different researchers.

Values of 2, 4, 6 and 8 were used on the simulated data, with the results shown in Figure 3.5. As can be seen, there is not a large difference between different settings. On average, using a value of 2 performs slightly better than the others; however, in practice this might be adjusted for the more important aim of convergence.

### 3.5.2 Fpeak

Another setting the researcher can change is *Fpeak*, a value that introduces rotations into PMF's computations. It is used based on trial and error by the researcher, and in environmental work "is not guaranteed to find the best solution." Positive values of *Fpeak* subtract rows from $\hat{\mathbf{F}}$ and add columns to $\hat{\mathbf{\Lambda}}$, while negative

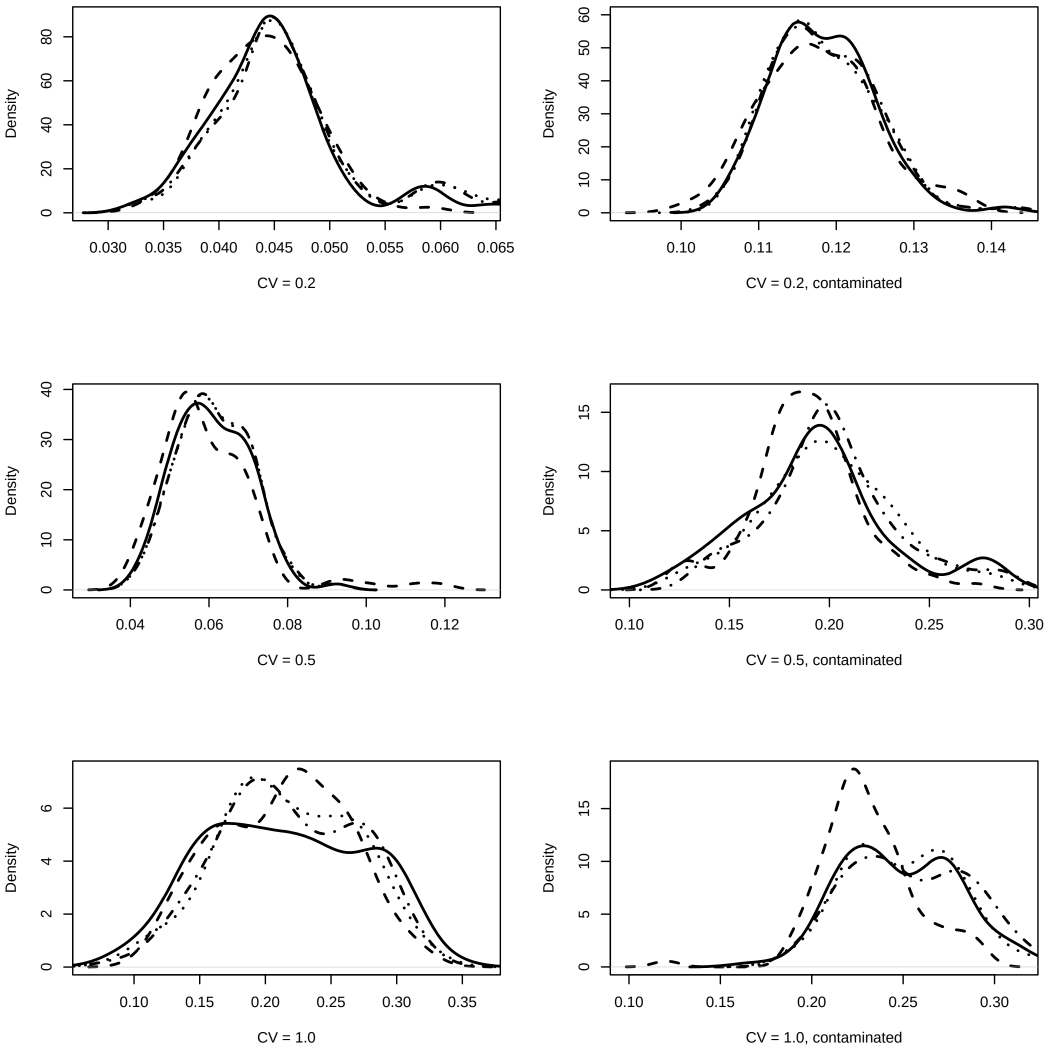


Figure 3.5: Kernel density estimates for the AAE of $\hat{\mathbf{F}}$ under different *outlier threshold distances*. Results for the default value of 4 are shown by the solid line, 2 by the dashed line, 6 by the dotted line, and 8 by the dash-dot line.

values do the opposite (Paatero, 2004a).

An example of the use of $Fpeak$ in practice is found in Hopke *et al.* (2004). Here, researchers use different values of $Fpeak$ that leave $Q$ (see Equation (3.3)) stable to explore rotational freedom in an analysis of data from Seattle, Washington. They use use these different results to gain an idea of the distribution of factor elements from PMF.

The simulation study results can be found in Figures 3.6, and 3.7. As can be seen, no value of $Fpeak$ greatly improves estimation of $\mathbf{\Lambda}$ over the default value of 0, while large positive and negative values can worsen estimation. In the estimation of $\mathbf{F}$, however, some values do lead to a marked improvement. With very clean data, (here, CV = 0.2, no extra sources), small negative values slightly improve estimation, while for data with more noise (CV = 1.0), large positive values can improve estimation.

For the practitioner, using a large positive value of $Fpeak$ could lead to improved estimation of source contributions, as long as the source profiles remain identifiable. A helpful step would be to calibrate actual airsheds with this study, to see if some can be identified as "cleaner" or "messier." As long as no method of calibration exists, using the default value of 0 seems to be the best alternative, since using these large positive values for $Fpeak$ sometimes improves estimation, but sometimes worsens it.

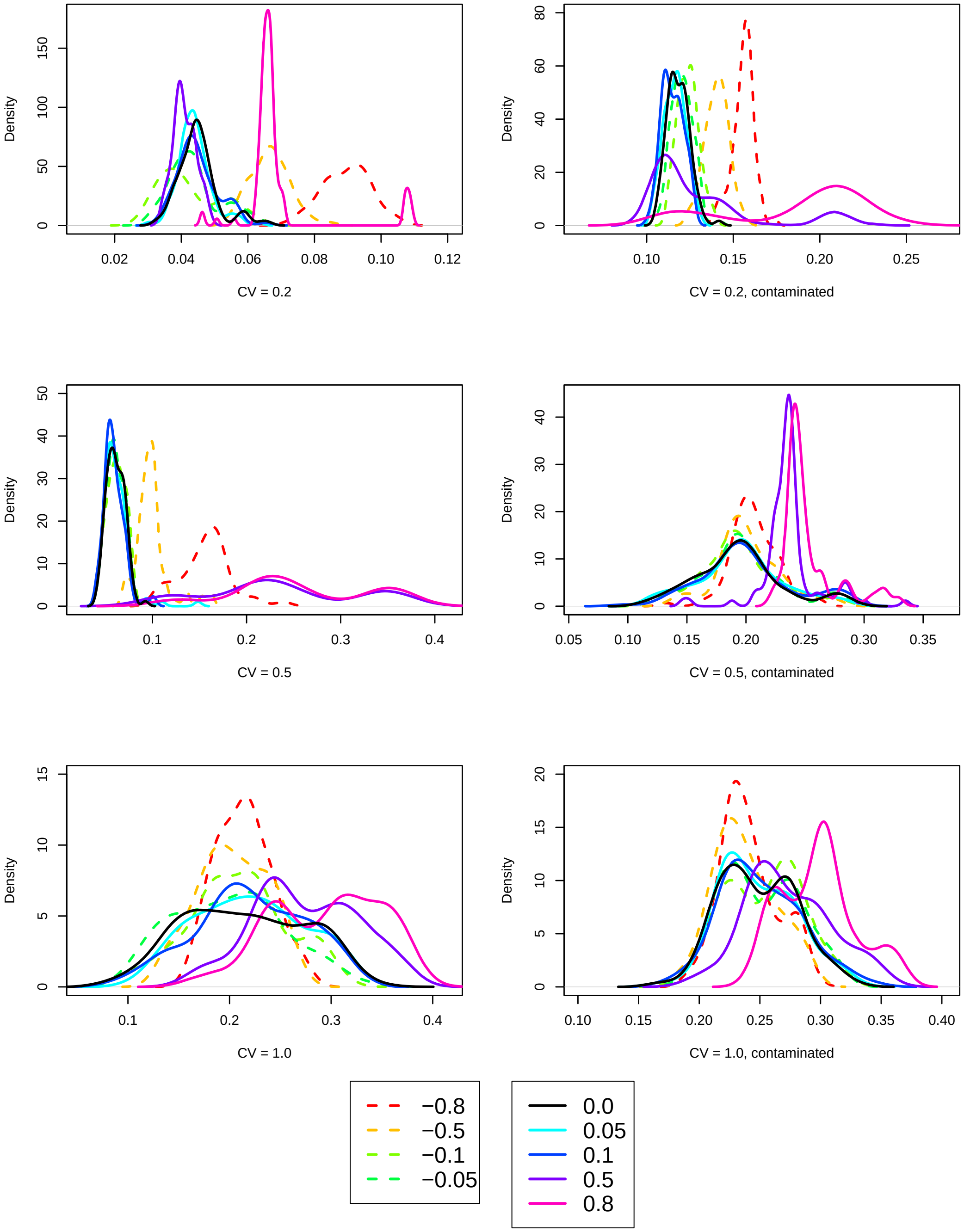


Figure 3.6: Kernel density estimates for the AAE of $\hat{\mathbf{\Lambda}}$ under different values of $Fpeak$. Results for the default value of 0.0 are shown by the solid black line, results for negative values by dashed lines, and positive values by solid colored lines.

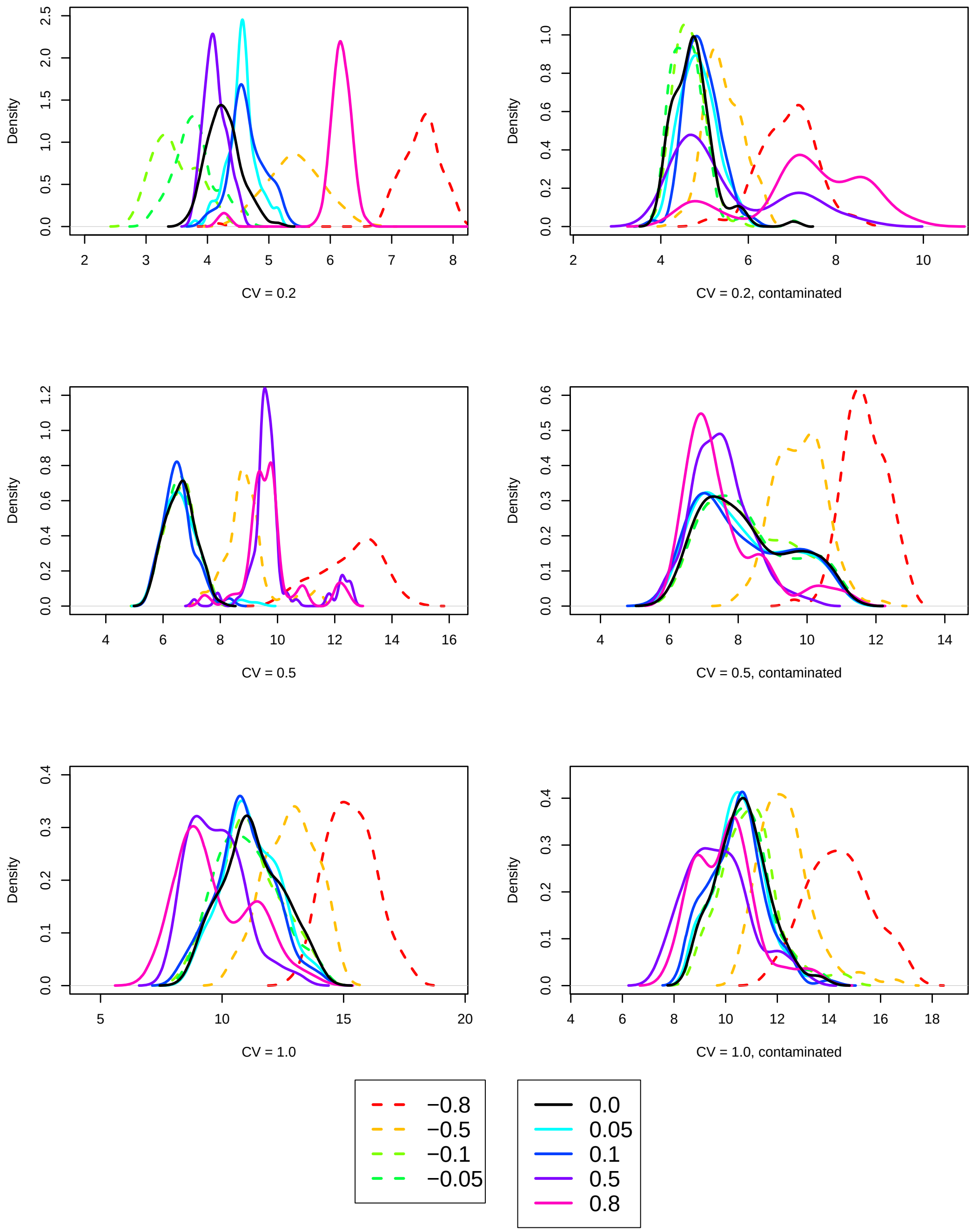


Figure 3.7: Kernel density estimates for the AAE of $\hat{\mathbf{F}}$ under different values of $Fpeak$. Results for the default value of 0.0 are shown by the solid black line, results for negative values by dashed lines, and positive values by solid colored lines.

### 3.5.3 Standard deviation computation

Four methods PMF uses to compute the standard deviations used in Equation (3.3) were compared. Each method uses a slightly different formula as shown below. In general, $t$ represents a measurement error, and $v$ represents "relative error." These equations and more details can be found in the PMF User's Guide, Part 2: Reference, pages 19-21 (2004).

**EM=-10** With the *EM*=-10 setting, PMF computes the standard deviations according to the formula

$$s_{ij} = \sqrt{t_{ij}^2 + .5v_{ij}^2|\hat{y}_{ij}|(|\hat{y}_{ij}| + |y_{ij}|)} \tag{3.5}$$

where $s_{ij}$ is the standard deviation used in Equation (3.3), $t_{ij}$ is the standard deviation of the measurement error, and $v_{ij}$ is the log-error standard deviation. The values $\hat{y}_{ij}$ and $y_{ij}$ represent the individual data points and their estimates, where $\hat{y}_{ij}$ is the fitted value obtained from $\hat{\mathbf{Y}} = \hat{\mathbf{\Lambda}}\hat{F}$, and $y_{ij}$ is the $ij$th observation from the data matrix $\mathbf{Y}$. PMF updates $s_{ij}$ as it recomputes its estimates, $\hat{y}_{ij}$, of $y_{ij}$. One can think of $s_{ij}$ as a modified version of the receptor measurement error, that is $s_{ij} = t_{ij}$ when $v_{ij} = 0$.

**EM=-12** Under the setting *EM*=-12, the standard deviations are computed in a simpler manner, according to Equation (3.6).

$$s_{ij} = t_{ij} + v_{ij}|y_{ij}| \tag{3.6}$$

The standard deviations are computed once and not changed as the program iterates. Again, when $v_{ij} = 0$, the standard deviation value $s_{ij}$ becomes the receptor measurement error.

**EM=-13** This setting iteratively computes $s_{ij}$ during fitting, but never uses the original data values.

**EM=-14** The final setting, *EM*=-14, that was evaluated computes the standard deviations in a manner similar to Equation (3.6), but instead of using $|y_{ij}|$, the program uses the maximum of $|y_{ij}|$ or $|\hat{y}_{ij}|$. This is the setting recommended for general environmental work.

#### 3.5.3.1 Input of $t_{ij}$ and $v_{ij}$

PMF can read $t_{ij}$ and $v_{ij}$ as arrays specified by the user, $\mathbf{T}$ or $\mathbf{V}$, or as constants *C1* and *C3* read in from the "ini" file. If the array input is used the equivalent constant is ignored.

If the user specifies no arrays for the input of standard deviations, PMF computes the standard errors according to an *ad hoc* formula Paatero (2004b). Also, one can see that when $v_{ij}$ is set equal to zero, the equations for *EM*=-10, -12, and -14 become equivalent.

#### 3.5.3.2 Study over wide range of settings

First, a study was performed on the contaminated data with CV = 0.2. A wide range of combinations of EM and values for C1 and C2 were tested. With regard to the AAE of $\hat{\mathbf{F}}$, PMF performed best when the array of standard deviations was read in as $\mathbf{T}$, using the values as $t_{ij}$ in the equations above. With $v_{ij}$ set equal to zero, all three methods returned the same results, an AAE of 4.84 for $\hat{\mathbf{F}}$ and an AAE of 0.117 for $\hat{\mathbf{\Lambda}}$. Values were computed as the mean of the data sets that ran. If PMF did not converge the results associated with each data set were not used, possibly leading to bias in the results. When a small value for $v_{ij}$ was introduced, the AAE for $\hat{\mathbf{F}}$ improved slightly, to 4.80, 4.70, and 4.78 for EM=-10, -12, -14, respectively.

Not specifying an array of standard deviations, that is using neither **T** or **V** as input, resulted in poor performance, with the AAE of $\hat{\mathbf{F}}$ ranging from 6.45 to 9.30.

From the performance of PMF when analyzing these data sets, the ideal setting seems to be either of the three methods of error computation, using **T** as the input array of standard deviations, and either 0 or a small value for $v_{ij}$ (read in from the "ini" file as *C3*). Partial results are shown in Table 3.3.

#### 3.5.3.3 Results

After the previous study, a smaller combination of *EM* and *C3* was tested over the entire range of simulated data, using **T** as the input of standard deviations. The results for this study are shown in Table 3.4 and Figure 3.8, which show the means, missing values omitted, of the results over 100 data sets for each combination. The most notable result is that using a large value for *C3* leads to significant increases in AAE for some data. However, with messy data (CV = 1.0), using a large value for *C3* leads to better results.

It seems that if the data is clean (all sources identified with a low CV), then the choice of *EM* and *C3* does not really matter— PMF succeeds in finding the correct results. With regard to **F**, as we increase the noise in the data, using a large value of

| $s_{ij}$ **computation** | **T or V** | **C1 (T)** | **C3 (V)** | $\hat{\mathbf{\Lambda}}$ **AAE** | $\hat{\mathbf{F}}$ **AAE** |
|---|---|---|---|---|---|
| 10 | T | 0 | 0 | 0.1165 | 4.8416 |
| 10 | T | 0 | 0.01 | 0.1177 | 4.7994 |
| 10 | T | 0 | 0.5 | 0.1067 | 7.0882 |
| 10 | V | 0.01 | 0 | 0.1678 | 7.5677 |
| 10 | V | 0.5 | 0 | 0.0856 | 7.3974 |
| 10 | Neither | 0.01 | 0 | 0.1638 | 8.8734 |
| 10 | Neither | 0.5 | 0 | 0.1638 | 9.2952 |
| 12 | T | 0 | 0 | 0.1165 | 4.8416 |
| 12 | T | 0 | 0.01 | 0.1162 | 4.6944 |
| 12 | T | 0 | 0.5 | 0.1090 | 6.3143 |
| 12 | V | 0.01 | 0 | 0.1676 | 7.2687 |
| 12 | V | 0.5 | 0 | 0.0860 | 7.3004 |
| 14 | T | 0 | 0 | 0.1165 | 4.8416 |
| 14 | T | 0 | 0.01 | 0.1168 | 4.7772 |
| 14 | T | 0 | 0.5 | 0.1146 | 6.8874 |
| 14 | Neither | 0 | 0.01 | 0.0833 | 7.0162 |
| 14 | Neither | 0 | 0.5 | 0.0843 | 7.1850 |

Table 3.3: Methods of standard deviation computation, study using CV = 0.2, contaminated data

*C3* seems to work as well or better than any other settings tested on the noisy data. To achieve optimal results, it appears that the researcher needs to have a sense of the amount of noise in the data and adjust the value of *C3* accordingly, when using $\mathbf{T}$ as the input array of standard deviations.

For estimation of $\mathbf{\Lambda}$, again with data where all sources are identified and the CV is 0.2 or 0.5, the different choices of *EM* and *C3* do not seem to matter— the noise in the data is low enough relative to the structure that PMF performs well in estimation regardless of the choice of settings. However, as the noise in the data increases, PMF performs significantly better using a large value of *C3*, allowing the program to better fit the large amount of error in the data. There seems to be some trade-off between estimation of $\mathbf{F}$ and $\mathbf{\Lambda}$, that as one improves the other might worsen.

If the researcher is more concerned with estimating source profiles instead of contributions, using a large value of *C3* seems to help, perhaps with some trade-off with estimating the contributions.

## 3.6 Conclusions

Positive Matrix Factorization is a valuable tool in the study of air pollution data. Once the mechanics of the program are mastered, good estimates are possible to find both quickly and easily.

In an analysis of real pollution data, the recommendation to the novice user would be to use the default settings, with adjustment of *C3* if they have a sense of the amount of noise in the data or are more interested in the source profiles. The *outlier*

**Source contributions**

| | **CV = 0.2** | **0.2, cont** | **0.5** | **0.5, cont** | **1.0** | **1.0, cont** |
|---|---|---|---|---|---|---|
| **10, .01** | 4.314 | 4.674 | 6.672 | 8.652 | 11.350 | 10.805 |
| **10, .1** | 4.207 | 4.956 | 6.644 | 7.896 | 11.313 | 10.876 |
| **10, .5** | 4.421 | 7.261 | 6.720 | 8.825 | 10.566 | 10.575 |
| **12, 0.0** | 4.259 | 4.833 | 6.655 | 8.654 | 11.449 | 10.821 |
| **12, .01** | 4.266 | 4.730 | 6.600 | 8.263 | 11.261 | 10.648 |
| **12, 0.1** | 4.348 | 4.839 | 6.402 | 7.260 | 10.333 | 9.772 |
| **12, 0.5** | 4.521 | 6.811 | 6.971 | 8.091 | 9.799 | 10.079 |
| **13, 0.01** | 4.312 | 4.727 | 6.663 | 8.491 | 11.487 | 11.011 |
| **13, 0.1** | 3.928 | 5.377 | 6.731 | 7.981 | 11.451 | 10.673 |
| **13, 0.5** | 4.622 | 7.950 | 6.848 | 8.927 | 11.613 | 11.730 |
| **14, .01** | 4.304 | 4.678 | 6.651 | 8.449 | 11.142 | 10.885 |
| **14, .1** | 4.305 | 4.989 | 6.473 | 7.602 | 10.238 | 10.178 |
| **14, .5** | 4.303 | 7.122 | 6.747 | 8.249 | 9.840 | 9.854 |

**Source profiles**

| | **CV = 0.2** | **0.2, cont** | **0.5** | **0.5, cont** | **1.0** | **1.0, cont** |
|---|---|---|---|---|---|---|
| **10, .01** | 0.046 | 0.118 | 0.062 | 0.206 | 0.219 | 0.249 |
| **10, .1** | 0.044 | 0.122 | 0.062 | 0.164 | 0.203 | 0.242 |
| **10, .5** | 0.048 | 0.104 | 0.063 | 0.130 | 0.132 | 0.204 |
| **12, 0.0** | 0.045 | 0.117 | 0.061 | 0.210 | 0.213 | 0.252 |
| **12, .01** | 0.045 | 0.118 | 0.061 | 0.194 | 0.215 | 0.248 |
| **12, 0.1** | 0.046 | 0.122 | 0.058 | 0.131 | 0.205 | 0.238 |
| **12, 0.5** | 0.047 | 0.101 | 0.062 | 0.136 | 0.177 | 0.222 |
| **13, 0.01** | 0.046 | 0.118 | 0.061 | 0.197 | 0.209 | 0.249 |
| **13, 0.1** | 0.040 | 0.127 | 0.062 | 0.152 | 0.188 | 0.226 |
| **13, 0.5** | 0.051 | 0.113 | 0.062 | 0.147 | 0.179 | 0.213 |
| **14, .01** | 0.045 | 0.118 | 0.061 | 0.198 | 0.210 | 0.252 |
| **14, .1** | 0.045 | 0.122 | 0.058 | 0.148 | 0.194 | 0.237 |
| **14, .5** | 0.047 | 0.106 | 0.061 | 0.137 | 0.191 | 0.217 |

Table 3.4: AAE for $\hat{\mathbf{F}}$ and $\hat{\mathbf{\Lambda}}$, using different combinations of *EM* and *C3*

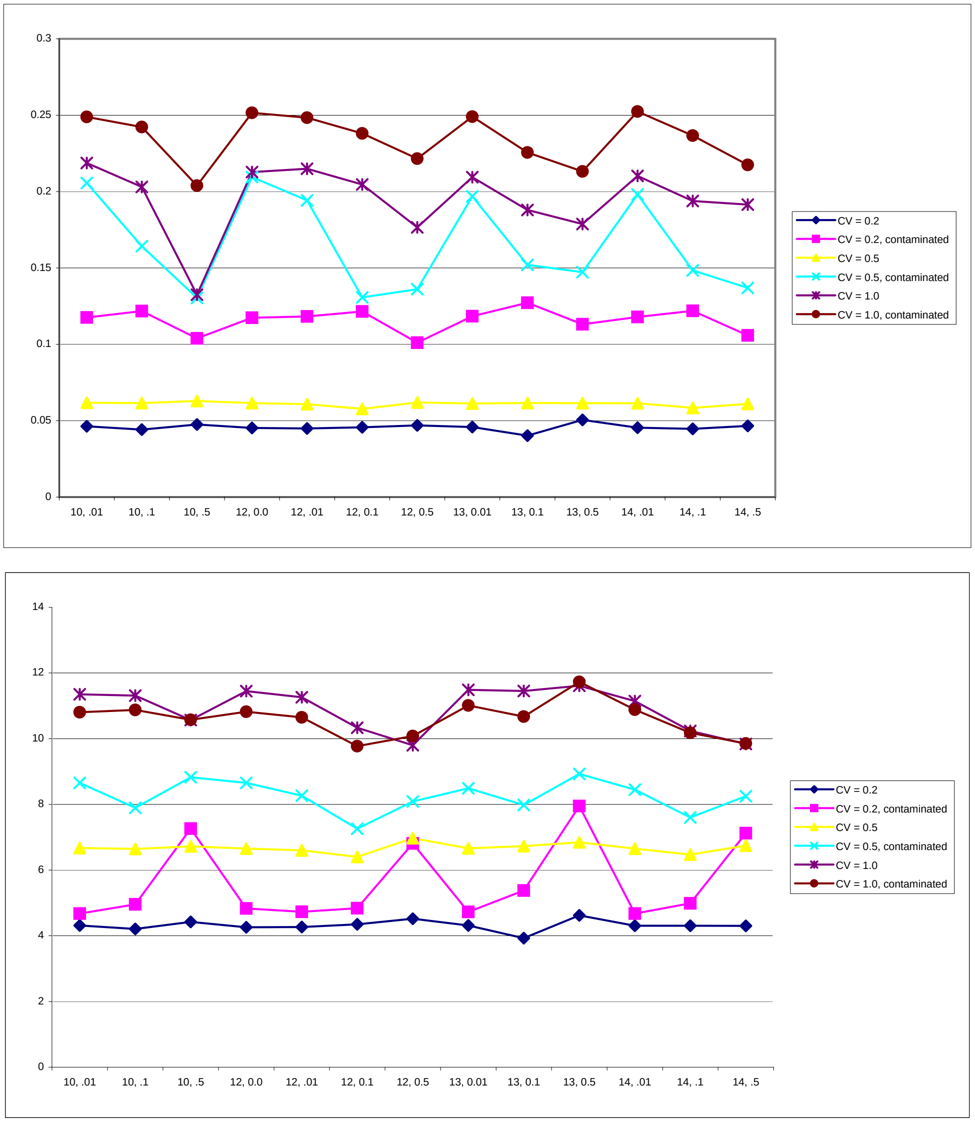


Figure 3.8: AAE for $\hat{\mathbf{\Lambda}}$ and $\hat{\mathbf{F}}$, using different combinations of *EM* and *C3*.

*threshold distance* might be adjusted to 2, if performance is not adversely affected in some other manner, and *Fpeak* is best left at zero.

# Chapter 4

# The use of *a priori* information in PMF

## 4.1 *A priori* information

In addition to the settings examined in Chapter 3 that modify its performance, PMF allows the user to introduce prior information into the algorithm. In this manner researchers can incorporate into the program knowledge from previous studies or facts known about a source believed to be present in the airshed. In theory, the introduction of correct information should improve the pollution source estimates. PMF offers the user two different ways to introduce this information, by pulling factor elements to zero and by using target factors. Both of these techniques in PMF use a "key" matrix, which give an integer value to each element of $\mathbf{\Lambda}$ or $\mathbf{F}$, and pulls that element to zero based on the integer value.

## 4.2 Pulling factor elements to zero ("Fkey")

Pulling factor elements to zero refers to the use of a matrix of *a priori* information for the zeros in source profiles or contributions. Since here we are concerned with the profiles, this information is given in the form of integer values in an *Fkey* (in

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Species** | Al | As | Br | Ca | Cl | Cu | EC | Fe |
| **Profile** | 0.00495 | 0.00032 | 0 | 0.03108 | 0.00735 | 0 | 0.17706 | 0.01616 |
| **Fkey** | 0 | 0 | 8 | 0 | 0 | 6 | 0 | 0 |
| **Species** | K | Mn | Na | Ni | NO3 | OC | P | Pb |
| **Profile** | 0.0076 | 0.00022 | 0 | 0.01422 | 0.00102 | 0.06379 | 0.00047 | 0.00163 |
| **Fkey** | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 |
| **Species** | Se | Si | SO4 | Sr | Ti | V | Zn | |
| **Profile** | 0 | 0.01342 | 0.63788 | 0 | 0.00057 | 0.01714 | 0.0051 | |
| **Fkey** | 7 | 0 | 0 | 8 | 0 | 0 | 0 | |

Table 4.1: Hypothetical elements of an *Fkey* for secondary sulfate.

the language of PMF, $\mathbf{F}$ is equivalent to our $\mathbf{\Lambda}$ and $\mathbf{G}$ is equivalent to our $\mathbf{F}$). The more sure the researcher is that a certain element of a source profile is zero, the larger the integer value that is specified. For example, in the profile for secondary sulfate used in the study, the elements of a hypothetical *Fkey* are shown in Table 4.1. The researcher in this example is quite sure that the concentration of Cu in the source is zero, and less sure about Br, Na, Se, and Sr.

Four different types of *Fkey* were used to examine the performance of this technique. First, 5 elements of the sea salt factor that are actually zero were pulled to zero with a value of 9, a “medium strong” pull. This represents a case where a researcher has knowledge about one of the factors, but not the others. Next, 10 elements chosen from the actual zeros scattered throughout $\mathbf{\Lambda}$ were pulled to zero with a value of 9, representing a case where the researcher has some knowledge about most of the sources. Third, 10 elements were again pulled to zero, this time with five pulled correctly to zero and five mistakenly pulled to zero, including one major element in the auto-diesel factor, as shown in Table 4.2. The fourth type of *Fkey*, like the second, again pulled 10 elements correctly to zero, this time with varying degrees of strength, from 5 to 14, to examine the effect of the $Fkey$ as we change the degree of certainty.

The AAE calculated using the first three $Fkeys$ is shown in Figures 4.1 ($\hat{\mathbf{F}}$) and 4.2 ($\hat{\mathbf{\Lambda}}$). As in Chapter 3, the resulting columns of $\hat{\mathbf{F}}$ and rows of $\hat{\mathbf{\Lambda}}$ were sorted based on a minimization of the MSE between columns of $\hat{\mathbf{F}}$ and columns of $\mathbf{F}$. As can be seen in Figures 4.1 and 4.2, when we correctly identify 5 zeros in the sea salt factor, we improve the overall estimation slightly in the clean data (CV = 0.2, and CV = 0.5 uncontaminated). When we correctly specify 10 profile zeros, we again improve estimation in the cleaner data. In the data that contains more noise, however, none of these three $Fkeys$ greatly change estimability. Perhaps estimation is so poor that correctly specifying zeros does little to change the results.

When we correctly specify five profile zeros, but incorrectly specify five others, as seen in the dotted line in Figures 4.1 and 4.2, we can dramatically worsen estimation in the cleaner data, while again leaving the noisier data estimates unchanged. Correctly pulling zero elements to zero can improve estimation, but if we misspecify the zeros, there might be a high price to pay.

Finally, the results from the fourth $Fkey$ are shown in Figure 4.3. Generally, for the cleaner data, increasing our level of certainty about the zeros improves estimation, while with noisier data the results are less certain, appearing to improve estimation in some situations and to worsen others. Interestingly, it appears that for source contributions, using a low degree of certainty may actually worsen estimation, even when correctly specifying profile zeros.

| | Sea salt | | Secondary Sulfate | | Winter Secondary | | Soil | | Auto-diesel | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Al** | 0 | 9 | 0.00495 | 0 | 0.03773 | 0 | 0.11493 | 0 | 0.00448 | 0 |
| **As** | 0 | 0 | 0.00032 | 0 | 0.00336 | 0 | 0 | 0 | 0 | 0 |
| **Br** | 0 | 0 | 0 | 0 | 0.00089 | 0 | 0 | 0 | 0.00006 | 0 |
| **Ca** | 0.009 | 0 | 0.03108 | 0 | 0.01546 | 0 | 0.00599 | 0 | 0.0075 | 0 |
| **Cl** | 0.265 | 0 | 0.00735 | 0 | 0.01921 | 0 | 0 | 0 | 0.00142 | 0 |
| **Cu** | 0 | 9 | 0 | 0 | 0.00658 | 0 | 0 | 0 | 0.00038 | 0 |
| **EC** | 0 | 0 | 0.17706 | 0 | 0 | 0 | 0.04891 | 0 | 0.3606 | 0 |
| **Fe** | 0 | 0 | 0.01616 | 0 | 0.04907 | 0 | 0.08181 | 0 | 0.00517 | 0 |
| **K** | 0.037 | 0 | 0.0076 | 0 | 0.00979 | 0 | 0.06708 | 0 | 0.00146 | 0 |
| **Mn** | 0 | 0 | 0.00022 | 0 | 0.0007 | 0 | 0.00125 | 9 | 0.00045 | 0 |
| **Na** | 0.689 | 0 | 0 | 0 | 0.05882 | 0 | 0.01565 | 0 | 0 | 0 |
| **Ni** | 0 | 0 | 0.01422 | 0 | 0.00605 | 0 | 0.00005 | 0 | 0 | 0 |
| **NO3** | 0 | 0 | 0.00102 | 0 | 0.28525 | 0 | 0 | 0 | 0.03768 | 0 |
| **OC** | 0 | 0 | 0.06379 | 0 | 0.02224 | 0 | 0.19562 | 0 | 0.52138 | 9 |
| **P** | 0 | 0 | 0.00047 | 0 | 0.00024 | 0 | 0 | 0 | 0.00069 | 0 |
| **Pb** | 0 | 0 | 0.00163 | 0 | 0.02884 | 0 | 0 | 0 | 0.00057 | 0 |
| **Se** | 0 | 0 | 0 | 9 | 0.00028 | 0 | 0 | 0 | 0 | 0 |
| **Si** | 0 | 0 | 0.01342 | 9 | 0.15755 | 0 | 0.45046 | 0 | 0.02284 | 0 |
| **SO4** | 0 | 0 | 0.63788 | 0 | 0.2767 | 0 | 0 | 0 | 0.03368 | 0 |
| **Sr** | 0 | 0 | 0 | 0 | 0.00083 | 0 | 0.00186 | 0 | 0.00006 | 0 |
| **Ti** | 0 | 0 | 0.00057 | 0 | 0.00608 | 9 | 0.01622 | 0 | 0.00026 | 0 |
| **V** | 0 | 9 | 0.01714 | 0 | 0.00758 | 0 | 0.00017 | 0 | 0 | 9 |
| **Zn** | 0 | 0 | 0.0051 | 9 | 0.00676 | 0 | 0 | 0 | 0.00131 | 0 |

Table 4.2: Third *Fkey*, representing correct and erroneous information about source profile zeros.

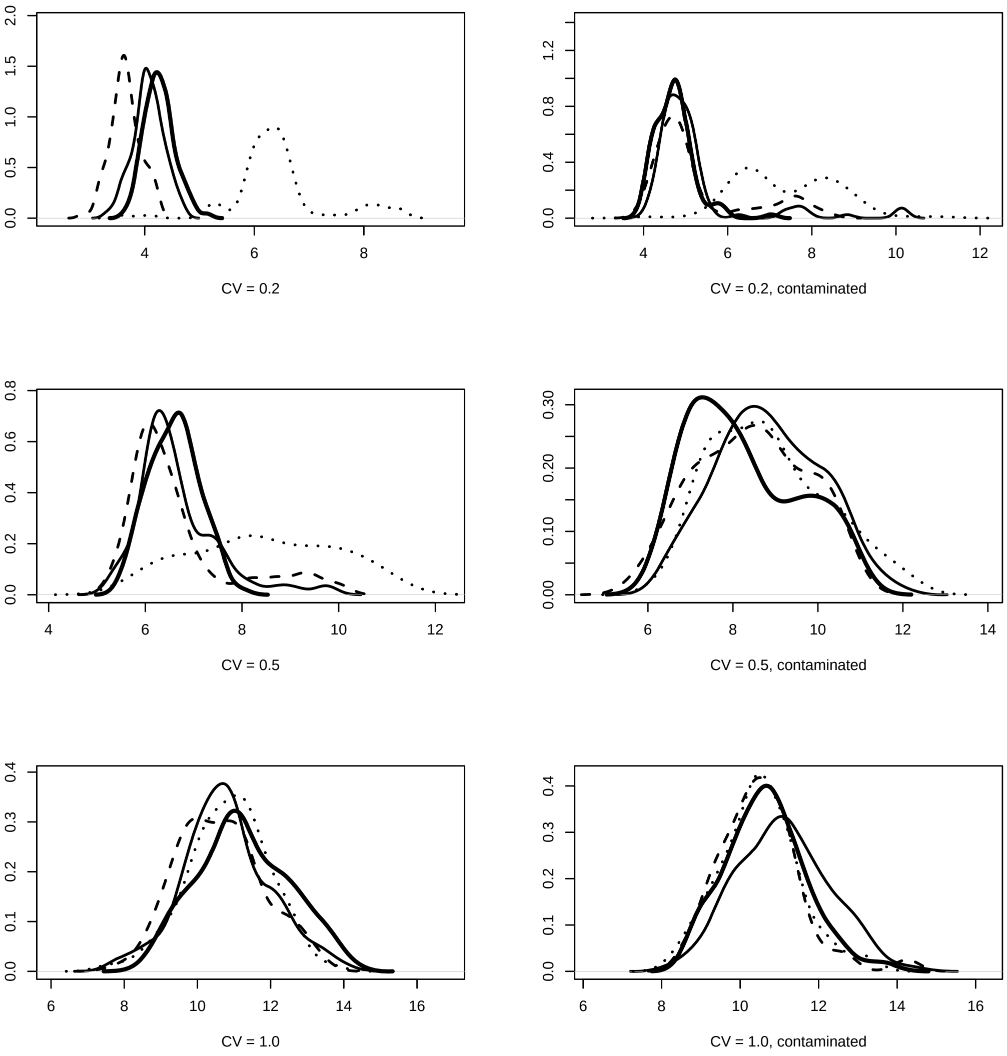
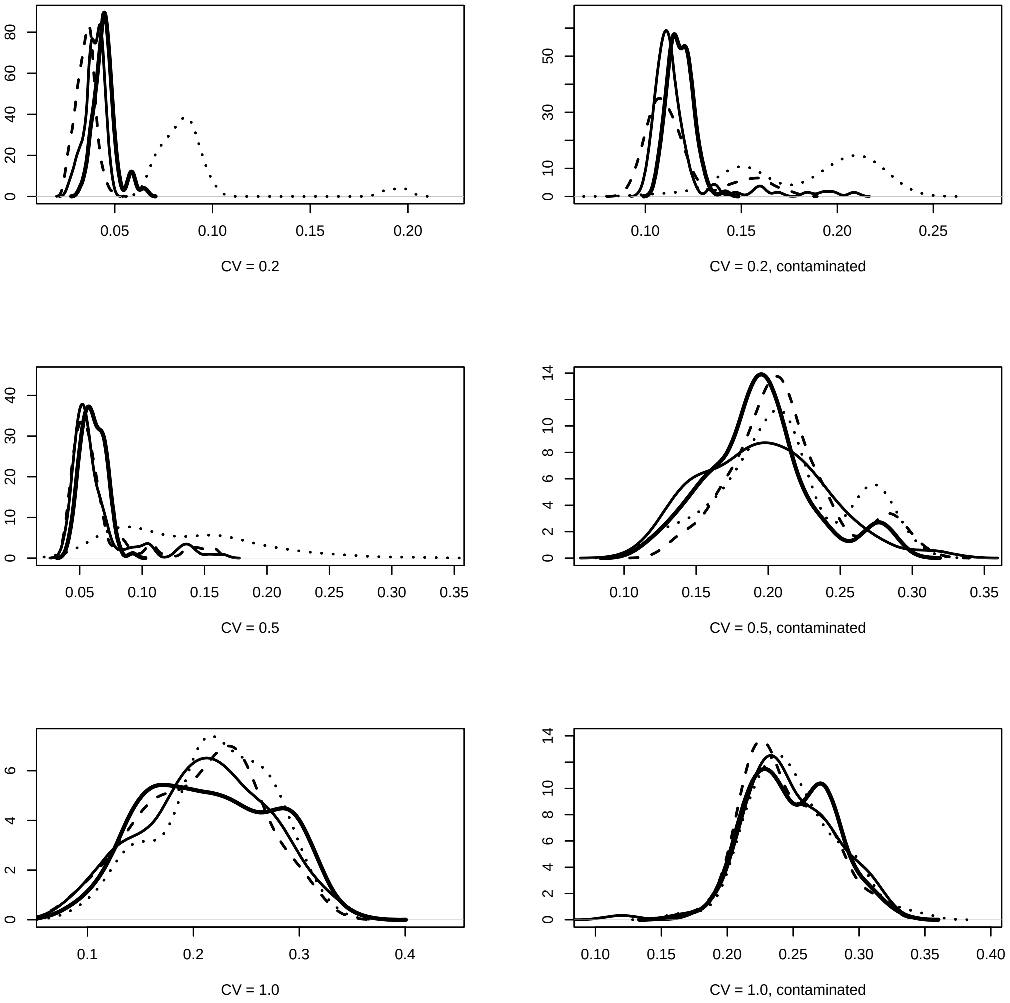


Figure 4.1: Kernel density estimates of the AAE for $\hat{\mathbf{F}}$, using three different $Fkeys$. The thick black line represents PMF's default results, the thin black line represents correctly pulling 5 elements of one factor to zero, the dashed line represents pulling 10 elements of multiple factors correctly to zero, and the dotted line represents a mixture of correct and incorrect pulling.



CV = 0.2

CV = 0.2, contaminated

CV = 0.5

CV = 0.5, contaminated

CV = 1.0

CV = 1.0, contaminated

Figure 4.2: Kernel density estimates of the AAE for $\hat{\mathbf{\Lambda}}$, using three different $Fkeys$. The thick black line represents PMF's default results, the thin black line represents correctly pulling 5 elements of one factor to zero, the dashed line represents pulling 10 elements of multiple factors correctly to zero, and the dotted line represents a mixture of correct and incorrect pulling.

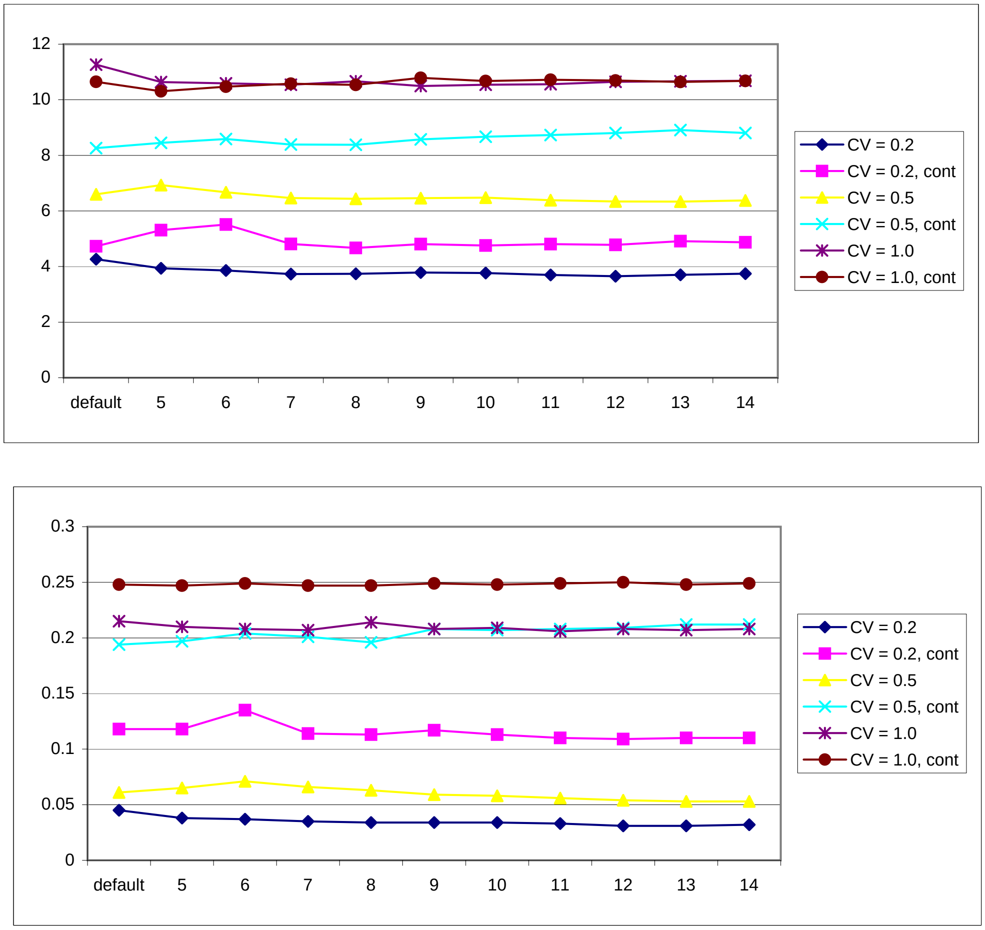


Figure 4.3: Means of the AAE results for the fourth *Fkey*, pulling 10 elements correctly to zero with differing amounts of strength.

## 4.3 Target source profiles ("Gkey")

The second method available for introducing prior information into the program is the use of target source profiles. This method introduces *a priori* information about one or more of the profiles in the form of a matrix of target profiles, $\tilde{\mathbf{\Lambda}}$. Each element of $\tilde{\mathbf{\Lambda}}$ is given a specified amount of uncertainty.

A matrix of estimated factor profiles was used as $\tilde{\mathbf{\Lambda}}$, obtained in the following manner. Let $\lambda_{ih}$ be the proportion contribution of the $i$th species to the $h$th profile, and let $\tilde{\lambda}_{ih}$ be its assumed value. We obtain $\tilde{\lambda}_{ih}$ as a draw from a lognormal distribution with a mean equal to $\lambda_{ih}$ and a coefficient of variation equal to either 100%, 50%, or 20%. Additionally, with each different CV, an array of uncertainties was passed to PMF. This array was obtained by multiplying $\tilde{\mathbf{\Lambda}}$ by 0.2.

As in the study of pulling elements to zero, once PMF had computed $\hat{\mathbf{\Lambda}}$ and $\hat{\mathbf{F}}$, the results were again sorted based on minimization of the MSE between $\hat{\mathbf{F}}$ and $\mathbf{F}$. If PMF failed to converge using the target profiles, the results from runs using the default values of PMF were substituted, where PMF had basically the same settings except for the lack of the *Gkey*. This corrects for the possibility that PMF is not converging on poorer data sets, although in practice the settings might be changed for a particular data set so that PMF converges even with the target profiles.

To use this *a priori* information with PMF, the following steps are used. First, $\tilde{\mathbf{\Lambda}}$ is appended to the beginning of the data matrix, and the $p \times k$ matrix containing the uncertainties associated with $\tilde{\mathbf{\Lambda}}$ is appended to the beginning of the matrix of receptor measurement uncertainties, so that now $\mathbf{Y}$ is $p \times (n+k)$, and our resulting $\hat{\mathbf{F}}$ will be $k \times (n+k)$. Next, a *Gkey* is made, which is constructed to pull these extra $k \times k$ elements of $\hat{\mathbf{F}}$ to zero. For details of the construction of this *Gkey*, see the PMF User's Guide, Part 1: Tutorial, page 16 (Paatero, 2004a).

Next, starting values are obtained from running PMF without the use of the target profile, and the results are used as starting points for the algorithm, after appending a $k \times k$ matrix of zeros to the beginning of the $\hat{\mathbf{F}}$ obtained from the run.

### 4.3.1 Starting values

Since the use of starting values complicates the process, it is desirable to see if the use of starting values actually improves the results. To test this, PMF was run over the 600 simulated data sets, using target profile information, with and without starting values. For this study, the true source profiles were used as $\tilde{\mathbf{\Lambda}}$, with the uncertainties associated with them obtained by simply multiplying the true source profiles by 0.2.

Figures 4.4 and 4.5 show the results from this simulation study, giving kernel density estimates of the distribution of the AAE associated with $\hat{\mathbf{F}}$ and $\hat{\mathbf{\Lambda}}$, respectively. Generally, the use of starting values improves results, more so for the data with a low CV.

Regardless of using starting values or not, except in the case of very clean data (CV = 0.2, all sources identified), the use of target source profiles for contaminated data does not dramatically improve estimation. It seems that unless we can correctly identify all the sources contributing to the particulate matter in the airshed, the use of target source profiles does not greatly improve our estimation.

### 4.3.2 Differing amounts of information in $\tilde{\mathbf{\Lambda}}$

We next consider how the use of target source profiles improves estimation when we have varying amounts of *a priori* information. This section examines the issue by using approximate profiles with varying degrees of uncertainty, specifically the $\tilde{\mathbf{\Lambda}}$'s are obtained as draws from lognormal distributions as described in the previous section. The degree of our knowledge is reflected by the level of the CV used to obtain $\tilde{\mathbf{\Lambda}}$. Thus, a CV of 0.2 reflects greater information than the CV of 1.0.

Three hundred different $\tilde{\mathbf{\Lambda}}$'s were thus created, one hundred each at CV = 0.25, 0.5, and 1.0. Each of these was used as prior information for six data sets, one for each different type of data (see Table 3.2 on page 17).

The AAE for $\hat{\mathbf{\Lambda}}$ and $\hat{\mathbf{F}}$ are shown in Figures 4.6 and 4.7, respectively, where kernel density estimates for the AAE associated with $\tilde{\mathbf{\Lambda}}_{CV=0.2}$, $\tilde{\mathbf{\Lambda}}_{CV=0.5}$, $\tilde{\mathbf{\Lambda}}_{CV=1.0}$ are shown in red, blue, and green, respectively. The default values, obtained without using *a priori* information, are shown by the solid black line, and the dashed black line shows the kernel density estimates obtained by using exact *a priori* information for comparison.

As can be seen in these figures, using the less informative $\tilde{\mathbf{\Lambda}}$ still improves the results, much in the same way as using exactly correct information. When we have uncontaminated data, or have correctly identified all the pollution sources, use of the

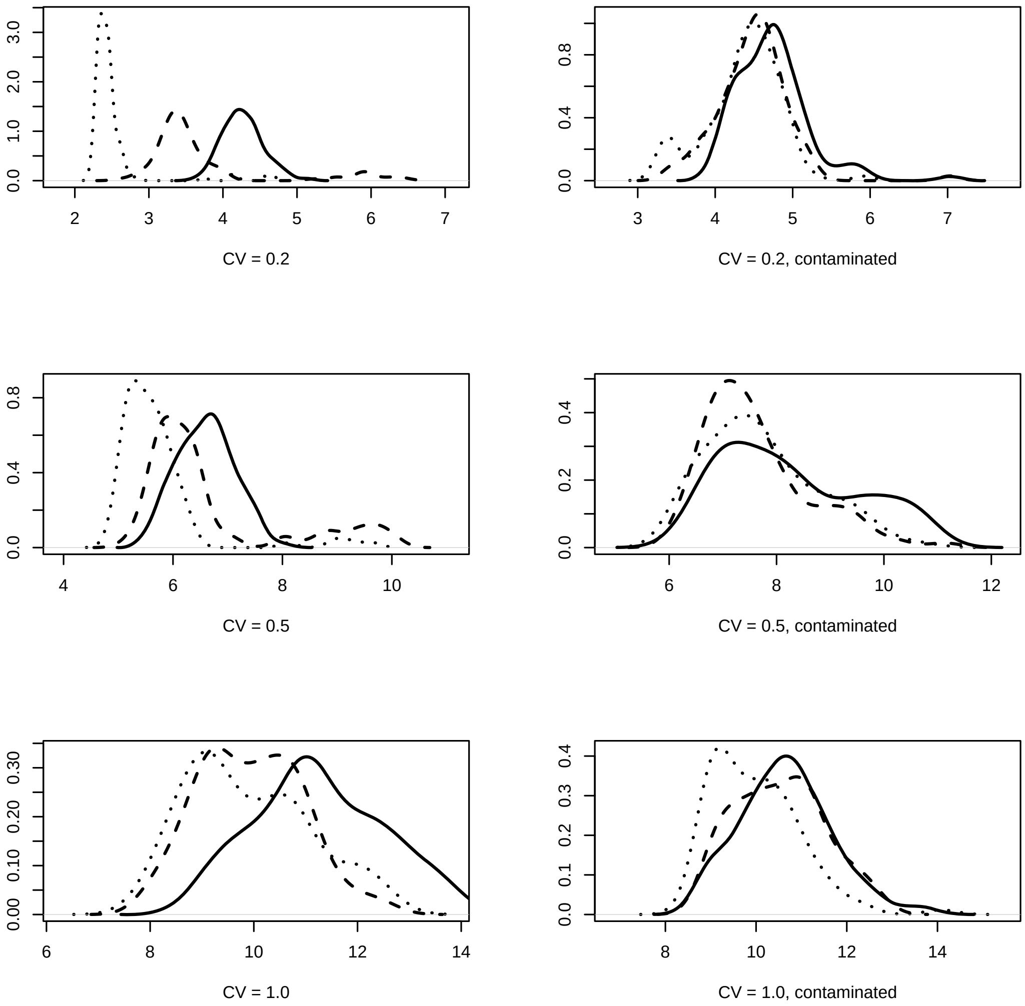


Figure 4.4: Kernel density estimates of the AAE for $\hat{\mathbf{F}}$, using target profiles in PMF. The solid black lines are the densities without target profiles, the dotted lines are the results from the use of target profiles with starting values, and the dashed lines show the results from the use of target profiles without starting values.

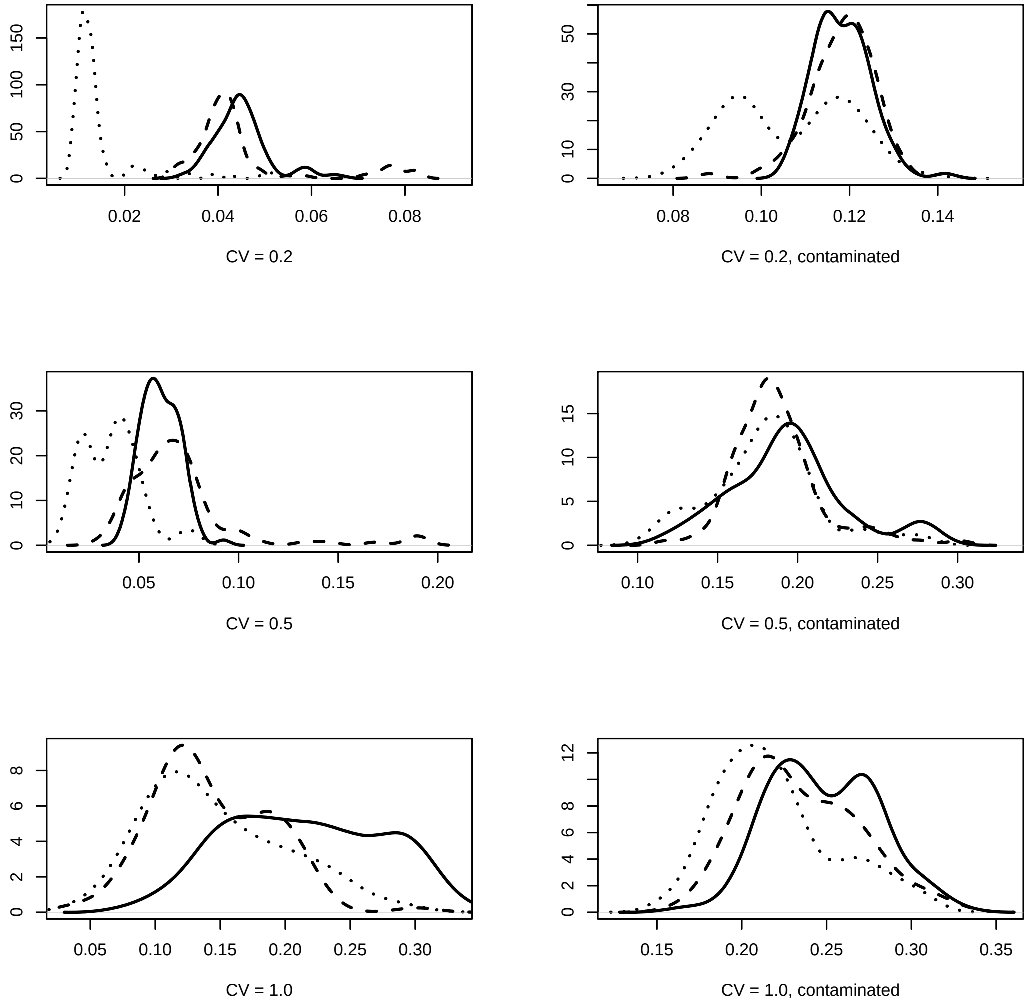


Figure 4.5: Kernel density estimates of the AAE for $\hat{\mathbf{\Lambda}}$, using target profiles in PMF. The solid black lines are the densities without target profiles, the dotted lines are the results from the use of target profiles with starting values obtained from an initial run of PMF, and the dashed lines show the results from the use of target profiles without starting values.

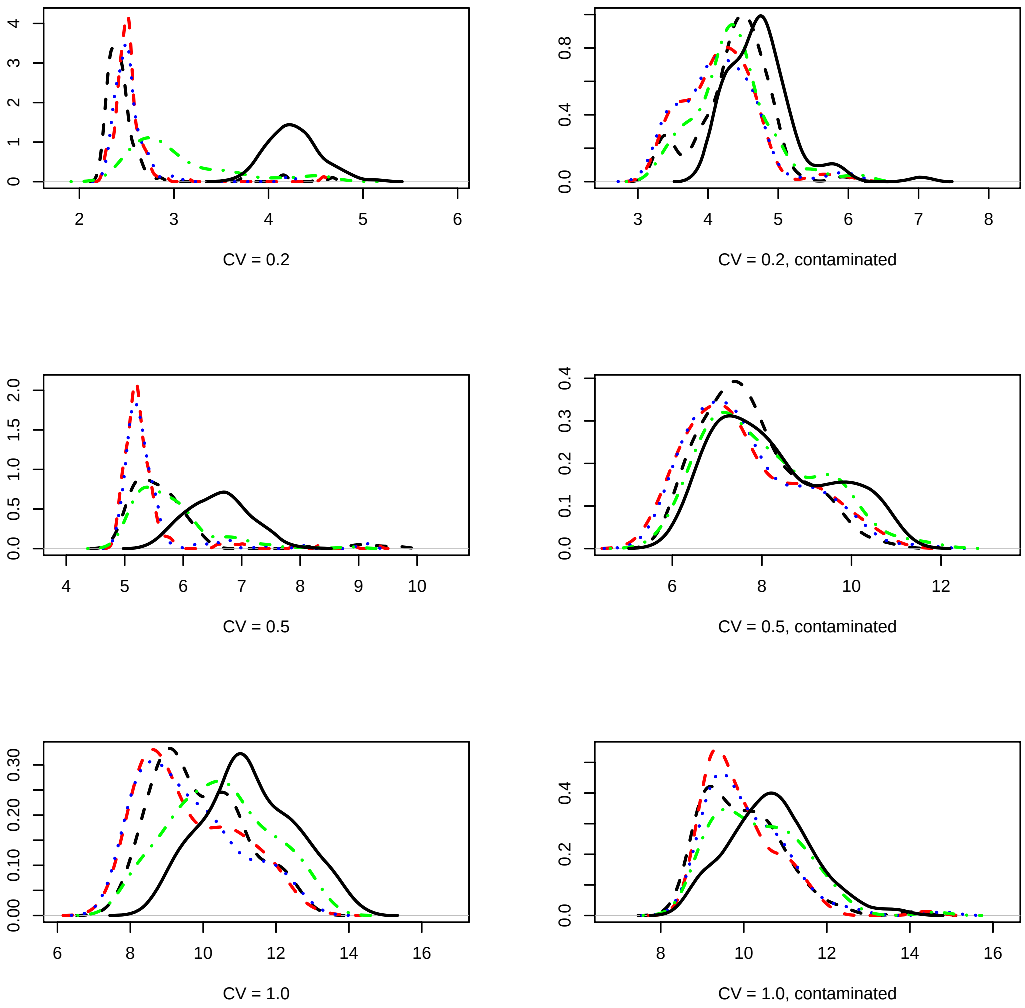
*a priori* source profiles improves estimation, less so for the least informative $\tilde{\mathbf{\Lambda}}$ (CV = 1.0). When we have not correctly identified all of the pollution sources, use of the *a priori* source profiles may slightly improve results for $\hat{\mathbf{F}}$. However, even when we have contaminated data, use of the *a priori* information can improve estimation of $\mathbf{\Lambda}$, even dramatically so when the data is clean (CV = 0.2, contaminated).

It appears that source profiles need not be exact for use of the *Gkey* to improve estimation. In fact, the use of exact information and good information behave similarly. The data seem to have enough structure that good estimates can be found with help that is less than exact. On the other hand, if we fail to correctly identify all the sources in the airshed, when the airshed is contaminated by unidentified sources, giving PMF good information about the sources we *do* know does not dramatically improve estimation.

In practice, the use of $Q$ from Equation (3.3) can give some indication as to when our *a priori* information has distorted the true shapes of factors, but since in this simulation study we know the true factors, AAE was used as the judge of estimation.

### 4.4 Illustration of use on the St. Louis airshed

In practice, the use of *a priori* information can be used to identify and constrain a model as we try to form a interpretable model of the airshed. We might have little idea, initially, of what sources are present, or how many. We can fit different numbers of sources in PMF, attempt to identify them, and then constrain those identifiable sources to be in the model, resulting in a model with a parsimonious number of



CV = 0.2

CV = 0.2, contaminated

CV = 0.5

CV = 0.5, contaminated

CV = 1.0

CV = 1.0, contaminated

Figure 4.6: Kernel density estimates of the AAE for $\hat{\mathbf{F}}$, using target profiles in PMF. Colored lines represent results from $\tilde{\boldsymbol{\Lambda}}$'s obtained as draws from lognormal distributions with CV = 0.2 (red), CV = 0.5 (blue), and CV = 1.0 (green).

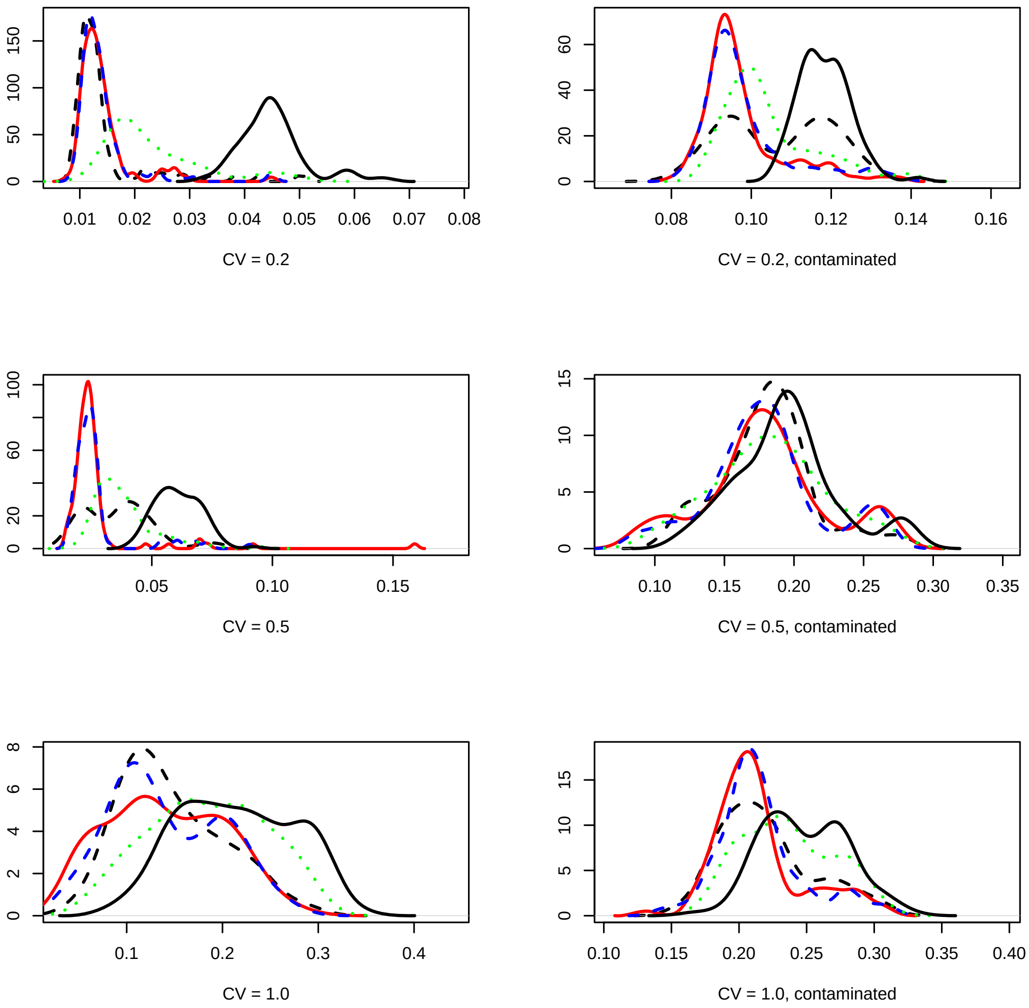


Figure 4.7: Kernel density estimates of the AAE for $\hat{\mathbf{\Lambda}}$, using target profiles in PMF. Colored lines represent results from $\tilde{\mathbf{\Lambda}}$'s obtained as draws from lognormal distributions with CV = 0.2 (red), CV = 0.5 (blue), and CV = 1.0 (green).

sources where each can be identified.

Data from the St. Louis - Midwest Supersite in 2001 - 2003 was fit in PMF, using 44 chemical species. Different numbers of sources were fit and labeled. One source appeared in more than one model that spiked on July 4th in both 2001 and 2002. The source was high in potassium and identified as a fireworks source. Since this source was easily identifiable, we sought to include it in the model. Other sources, such as summer secondary, were also identifiable, some to a greater degree than others.

An additional tool that was employed in identifying the sources was the use of weekday plots, plotting the level of the source contribution over the days of the week. For example, the vehicular sources peak during the middle of the week and decrease over the weekend, which is what one might expect as people go to work during the week. Sources that are not man-made, like the secondary sources, exhibited less day-to-day predictability and more seasonal predictability.

We used a $\tilde{\mathbf{\Lambda}}$ constructed from the estimates of source profiles we could identify to constrain these sources to appear in the model. In this manner, we could help PMF identify sources that we knew should be included in the model. In the end, we fit a six source model, with a source for fireworks, summer secondary, winter secondary, a smelter, and two vehicular sources. This model is shown in Figure 4.8, with plots of the source contributions by days of the week in Figure 4.9. The use of the *Gkey* proved to be very important and useful in the process of model fitting.

Although the use of the *Gkey* and *a priori* information is complex, it is a valuable tool that can aid researchers working in pollution source apportionment.

Without the ability to constrain sources to be in the model as they are identified, the researcher using PMF would possibly have to accept sources they cannot identify in order to include sources in the model that they can identify. For example, a 10 source model might contain a known source but has other sources that cannot be labeled, while PMF's output for a five source model does not contain the known source. Using the *Gkey*, one can constrain the known source that appeared in the 10 source model to appear in the five source model, giving a model that has a parsimonious number of sources, all of which can be identified.

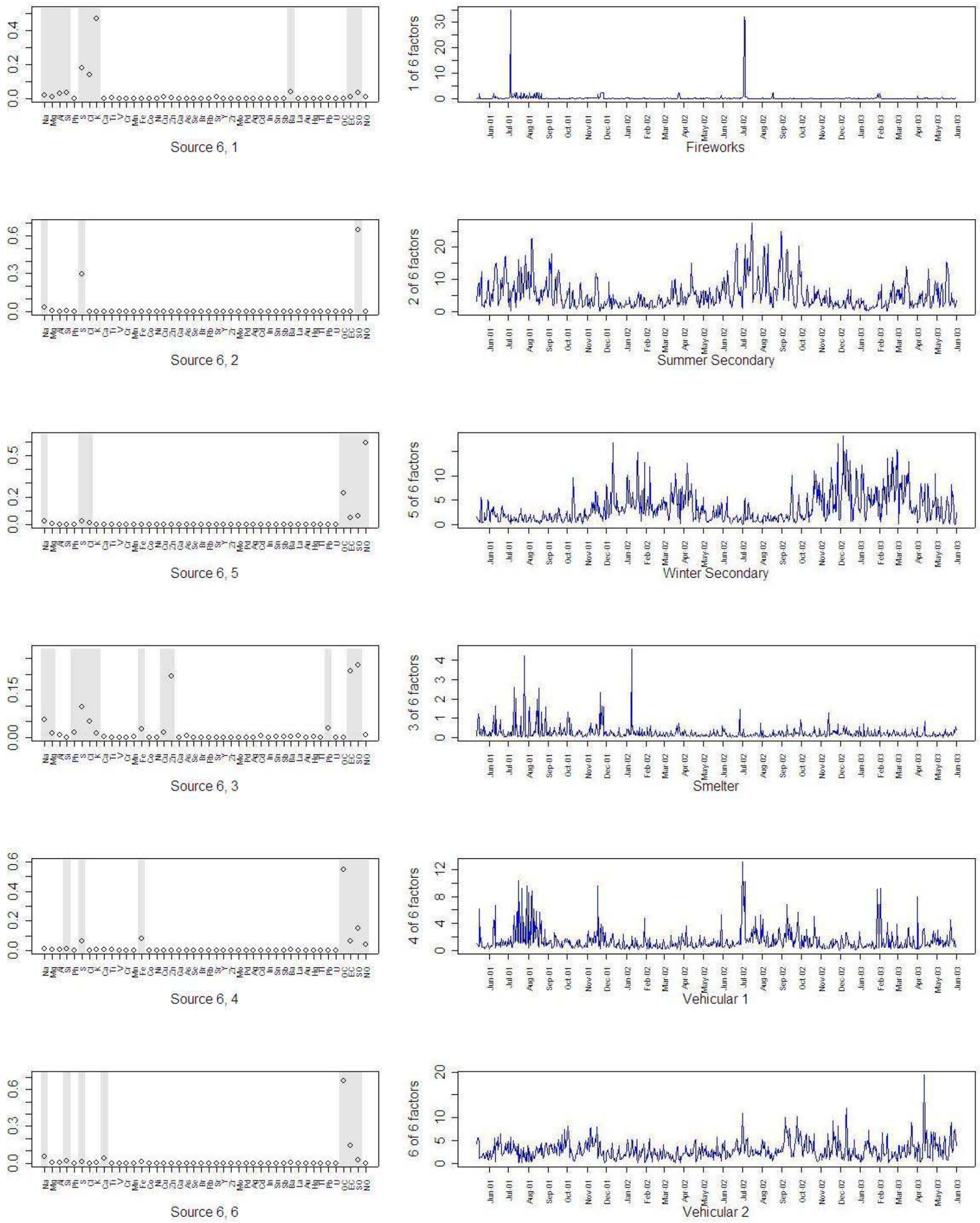


Figure 4.8: Six source model of the St. Louis airshed, using *a priori* information in PMF.

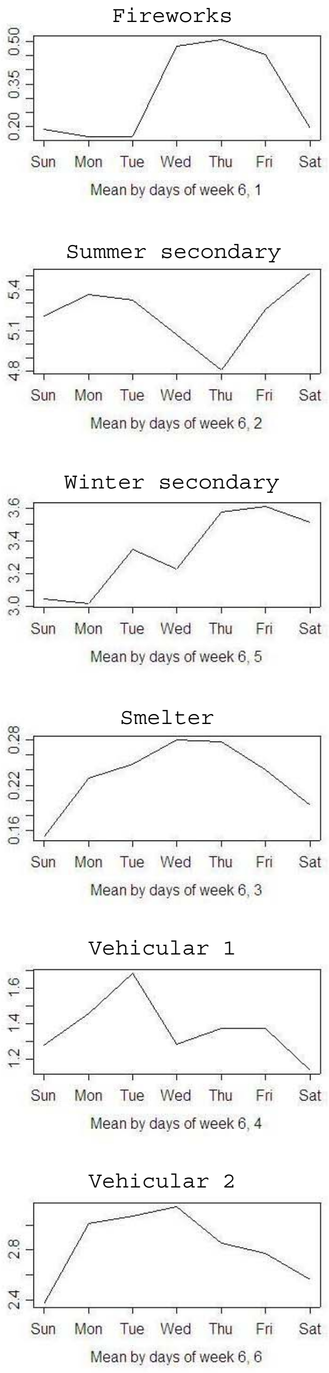


Figure 4.9: Six source model of the St. Louis airshed, using *a priori* information in PMF, by days of week.

# Chapter 5

## A Bayesian model for pollution source apportionment

This chapter proposes a Bayesian approach to pollution source apportionment, implemented in MATLAB and fitted using a Gibbs-Metropolis algorithm. The Bayesian model is desirable for multiple reasons. First, we obtain densities instead of point estimates for both source profiles and contributions, as in PMF. There are many advantages of having a density instead of a point estimate: we have an intrinsic measure of the uncertainty associated with the estimate and we can easily derive other quantities of interest such as credible intervals. PMF does afford some capability of finding distributional properties through multiple runs (Hopke *et al.*, 2004), but not in the intrinsic manner that Bayesian methods give.

Second, the incorporation of *a priori* information is natural in the Bayesian paradigm; we can simply use the knowledge we have about source profiles as the prior distributions on elements of $\mathbf{\Lambda}$. We can also use prior distributions on the elements of $\mathbf{F}$ reflecting our knowledge, if any, of the source contributions. In this chapter, the use of lognormal prior distributions is illustrated, building on the research of Park *et al.* (2001).

Third, the type of *a priori* information we can use in the estimation process is different from PMF. In PMF we give target profiles as points, each with a corresponding measure of certainty. In the Bayesian framework we specify distributions as the prior information, allowing us to incorporate more knowledge than a point estimate and measure of uncertainty.

Fourth, compared to PMF, the researcher has complete control over the process of estimation, instead of relying on the comparatively cryptic implementation of the PMF algorithm found in PMF2 (Paatero, 2004a) or the EPA implementation of PMF (Eberly, 2005). Some of the multiple settings of PMF were examined in Chapter 3, but even more remain. Although the PMF algorithm is explained, the complete process remains in the compiled program. In contrast, with this Bayesian framework the estimation process become much more open.

Fifth, posterior distributions have established statistical properties, which PMF lacks. We can look at results from a Bayesian model in terms of probability, whereas the results from PMF are harder to examine statistically. On the other hand, PMF is much quicker and easier to use. A PMF analysis can run in less than a minute on a normal computer, whereas fitting a high- dimensional Bayesian model may take days and require intense coding.

As compared to the Effective Variance solution, a Bayesian model allows source profiles that have been erroneously specified to correct themselves during the process of fitting the model. Additionally, allowing for non-Gaussian distributions in the model gives the researcher increased flexibility and the ability to incorporate more specific knowledge into the model.

## 5.1 Model

The model assumes that the interaction between pollution sources is additive, that $\boldsymbol{\mu}_j = \mathbf{\Lambda}\mathbf{f}_j$, where $\boldsymbol{\mu}_j = (\mu_{j1}, \mu_{j2}, \ldots, \mu_{jp})$ is a $p \times 1$ vector of the true ambient measurements on $p$ chemical species at time $j$, $\mathbf{\Lambda}$ is the $p \times k$ matrix of $k$ true profiles, and $\mathbf{f}_j$ is the $k \times 1$ vector of true source contributions of $k$ sources at time $j$.

Next, we assume that each of the $p$ ambient measurements at time $j$, $y_{ij},\ i = 1, \ldots, p, j = 1, \ldots, n$, has a lognormal distribution with mean $\mu_{ij}$ and some coefficient of variation (CV) $v$, which may easily be allowed to vary between chemical species. The lognormal distribution is convenient since it keeps all values greater than zero, while allowing for large positive values in the tail. This distribution makes sense scientifically, since we expect unusually large values due to the complex nature of an airshed. The lognormal density function is denoted by $\psi$ in $\psi(y_{ij}|\mu^*_{ij}, \tau^*_{ij}) = \frac{1}{\tau^*_{ij} y_{ij}\sqrt{2\pi}}\, e^{\frac{-(ln(y_{ij})-\mu^*_{ij})^2}{2\tau^{*\,2}_{ij}}}$, where $\mu^*_{ij}$ and $\tau^*_{ij}$ represent the parameters of the lognormal distribution in terms of the mean and CV ($\mu^*_{ij} = log(\mu_{ij}) - .5log(v^2+1)$, $\tau^*_{ij} = log(v^2+1)$), and $i = 1, \ldots, p\ ; j = 1, \ldots, n$.

For the prior distributions on elements of $\mathbf{\Lambda}$, we again use lognormal distributions, which will ensure non-negativity. Specifically, we assume that $\lambda_{ih}$ has a lognormal distribution with mean $\tilde{\lambda}_{ih}$ and coefficient of variation $u_{ih}$, where $\tilde{\lambda}_{ih}$ is the $ih$th element of a matrix of *a priori* source profiles, $\underset{p\times k}{\tilde{\mathbf{\Lambda}}}$ which has $\sum_{i=1}^{p} \tilde{\lambda}_{ih} = 1$. We obtain $u_{ih}$ in the following manner: if $\tilde{\lambda}_{ih}$ is greater than 0.4, set $u_{ih}$ equal to 0.02. If

$\tilde{\lambda}_{ih}$ is greater than 0.1 and less than 0.4, set $u_{ih}$ equal to 0.05. If $\tilde{\lambda}_{ih}$ is greater than 0.01 and less than 0.1, set $u_{ih}$ equal to 0.1. If $\tilde{\lambda}_{ih}$ is greater than 0.001 and less than 0.01, set $u_{ih}$ equal to 0.2. and, finally, if $\tilde{\lambda}_{ih}$ is less than 0.001, set $u_{ih}$ equal to 0.5. In this manner we reflect relatively greater certainty about the larger profile elements and relatively less certainty about those close to zero. The prior distributions for some elements of the winter secondary source are shown in Figure 5.1.

For the prior distribution on elements of $\mathbf{F}$, we take each element $f_{hj} \sim \psi(\gamma, \delta)$, using the same prior distribution for each element, reflecting a low level of knowledge about the source contributions. The lognormal distribution again keeps elements of $\mathbf{F}$ positive while allowing for large contributions of source $h$ to the airshed. Using $\gamma = 2$ and $\delta = 1$, we obtain the relatively diffuse distribution shown in Figure 5.2.

Equation (5.1) shows the un-normalized posterior distribution for $\mathbf{\Lambda}$ and $\mathbf{F}$ given $\mathbf{Y}$, where $\boldsymbol{\lambda}_i'$ is the $i$th row of $\mathbf{\Lambda}$, and $\mathbf{f}_j$ is the $j$th column of $\mathbf{F}$, corresponding to time $j$. We obtain draws from the normalized posterior by using Markov chain Monte Carlo methods. For interpretability, we force the columns of $\mathbf{\Lambda}$ to sum to one, assuming that the $p$ chemical species account for 100% of the emissions from the source. This is accomplished in computation by drawing candidate value $\lambda_{ih}^*$ as usual in the Metropolis algorithm, updating $\boldsymbol{\lambda}_h$ with this new $i$th element, and then scaling $\boldsymbol{\lambda}_h$ to sum to one. The proposed $\boldsymbol{\lambda}_h$ is then accepted or rejected by regular Metropolis steps.

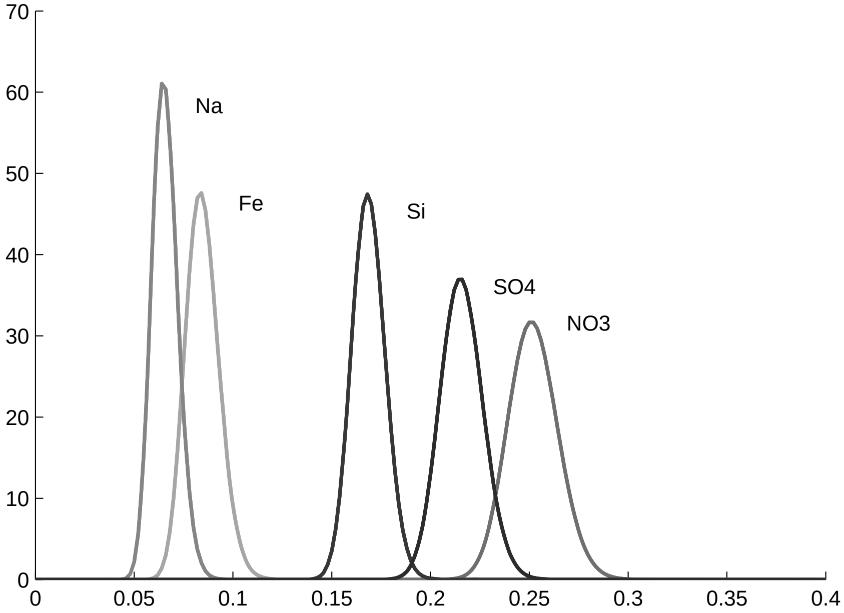


Figure 5.1: Lognormal prior distributions for the five largest elements of the winter secondary source

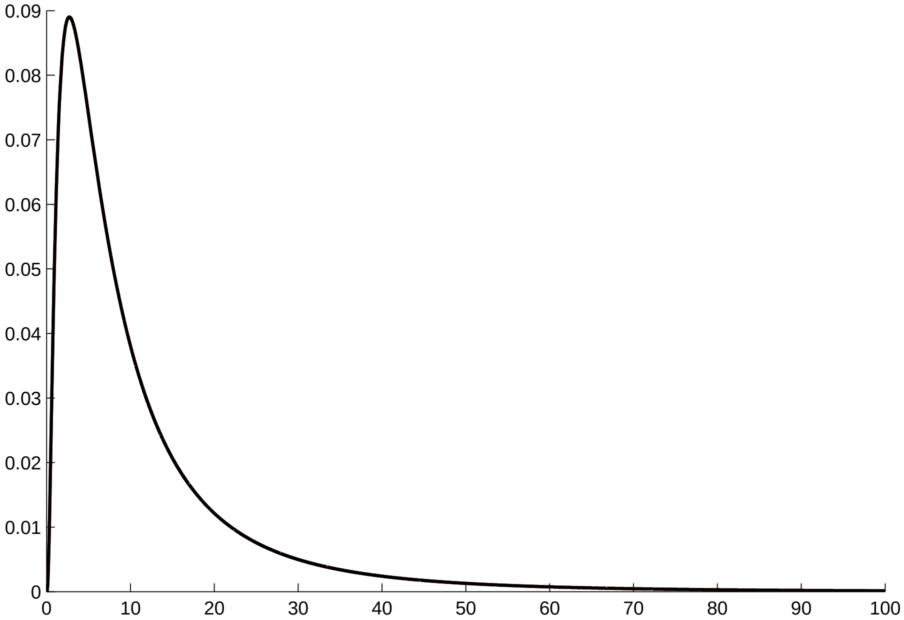


Figure 5.2: Lognormal prior distribution for the elements of **F**

$$p(\mathbf{\Lambda}, \mathbf{F}|\mathbf{Y}) \propto \prod_{j=1}^{n}\prod_{i=1}^{p}\prod_{h=1}^{k} \psi(y_{ij}|\boldsymbol{\lambda}_i\mathbf{f}_j, CV = v)\ \psi(f_{hj}|\gamma, \delta)\ \psi(\lambda_{ih}|\tilde{\lambda}_{ih}, CV = u_{ih}) \quad (5.1)$$

## 5.2 Computation

The data used in the study were simulated as described in Chapter 3, using contaminated data (i.e. minor sources were included in the airshed but not in the model) and a coefficient of variation of 0.2. Draws from the normalized posterior were obtained by using an adaptive Gibbs-Metropolis algorithm, where normal distributions were used as the candidate distributions. Candidate distributions are the distributions from which updates of the parameters are drawn (proposed) to be accepted or rejected by the algorithm. In part, the success of the Metropolis algorithm depends on the candidate distributions chosen. If the candidate distributions are too diffuse, few points will be generated that are accepted by the algorithm and the parameter space will not be explored efficiently. Similarly, if the candidate distributions are too narrow, too many points will be accepted all in a small range, leading to slow exploration of the parameter space. Gelman *et al.* (2004) state rough guidelines.

> Increase or decrease the scale of the jumping distribution if the acceptance rate of the simulations is much too high or low, respectively. The goal is to bring the jumping rule toward the approximate optimal value of 0.44 (in one dimension) or 0.23 (when many parameters are being updated at once using vector jumping).

When dealing with a small number of parameters, the researcher may change the candidate distributions until good mixing is achieved. However, in this problem we are trying to estimate $788 \times 5 + 23 \times 5 = 4055$ parameters, so adjusting each by hand is difficult. Instead, the algorithm keeps track of acceptance ratios as the iterations advance during the burn-in phase. If too many proposed points are being accepted, the standard deviation of the candidate distribution for that element is raised, or if too few proposed points are being accepted, the standard deviation of the candidate distribution for that element is lowered so that more points in high-density areas are generated and accepted. Specifically, elements of $\mathbf{F}$ where acceptance ratios fell below 0.25 and elements of $\mathbf{\Lambda}$ where acceptance ratios fell below 0.20 had the standard deviations of their corresponding candidate distributions lowered by a tenth. The elements of $\mathbf{F}$ where acceptance ratios exceeded 0.55 and the elements of $\mathbf{\Lambda}$ where acceptance ratios exceeded 0.60 had the standard deviations of their corresponding candidate distributions increased by a tenth. This updating was performed every 100 iterations during the burn-in phase of the algorithm. This technique resulted in quite good simulation of the elements of the posterior density.

The simulations are performed in MATLAB, which is well-suited for fitting Bayesian models. MATLAB has a powerful matrix language, similar to that of R, but performs much faster. The ability to vectorize loops and perform computations simultaneously speeds the algorithm. A crucial element of speed is also to eliminate redundant calculations, which can be achieved by examining the algorithm and identifying elements of the density that are identical as we compare an update to its previous value. Since these identical values will cancel as we compare the density under the update and the current value, they need not be performed.

## 5.3 Results

A run of 100,000 iterations was performed, with the last 10,000 iterations being kept for examination. The final acceptance ratios for $\hat{\mathbf{F}}$ and the final adjusted candidate sigmas are shown in Figure 5.3. A similar plot for the elements of $\hat{\mathbf{\Lambda}}$ is shown in Figure 5.4. As can be seen, our acceptance ratios are all in a reasonable range, indicating that our algorithm to adjust the standard deviations of the candidate distributions worked and we achieved reasonably good mixing.

The trace plots for $\hat{\mathbf{\Lambda}}$ are shown in Figures 5.5 through 5.9. The solid red lines indicate the true values of $\mathbf{\Lambda}$ from which the data were simulated, the solid blue lines indicate the means of the prior distributions. The trace plots show reasonably well mixing. As can be seen, in many cases where the algorithm moves away from the prior distribution, the algorithm succeeds in finding the correct value, or in finding a point between the mean of the prior distribution and the correct value. In other cases, as in Figure 5.8, the algorithm moves away from the mean of the prior distribution in the wrong direction.

As examples of the trace plots for $\hat{\mathbf{F}}$. Figure 5.10 shows the trace plot and density estimates for $\hat{\mathbf{f}}_{200}$ and Figure 5.11 shows the trace plots and density estimates for $\hat{\mathbf{f}}_{400}$. The red lines indicate the values of $\mathbf{F}$ from which the data were simulated.

The AAE associated with $\hat{\mathbf{\Lambda}}$ ($\hat{\mathbf{\Lambda}}$ obtained from the posterior means) is 0.041, which is slightly worse than the AAE for the means of the prior distributions, 0.037. The AAE associated with $\hat{\mathbf{F}}$ ($\hat{\mathbf{F}}$ obtained from the posterior means) is 6.316.

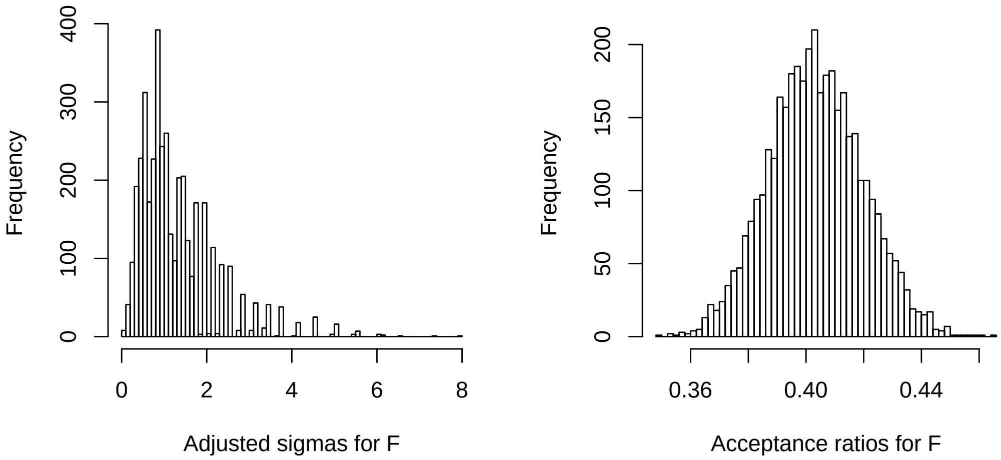


Figure 5.3: Final candidate sigmas and acceptance ratios for $\hat{\mathbf{F}}$

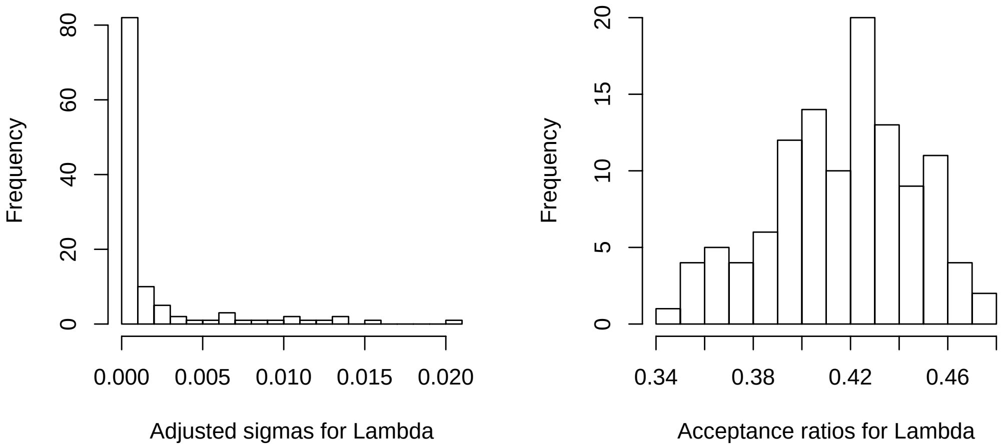


Figure 5.4: Final candidate sigmas and acceptance ratios for $\hat{\mathbf{\Lambda}}$

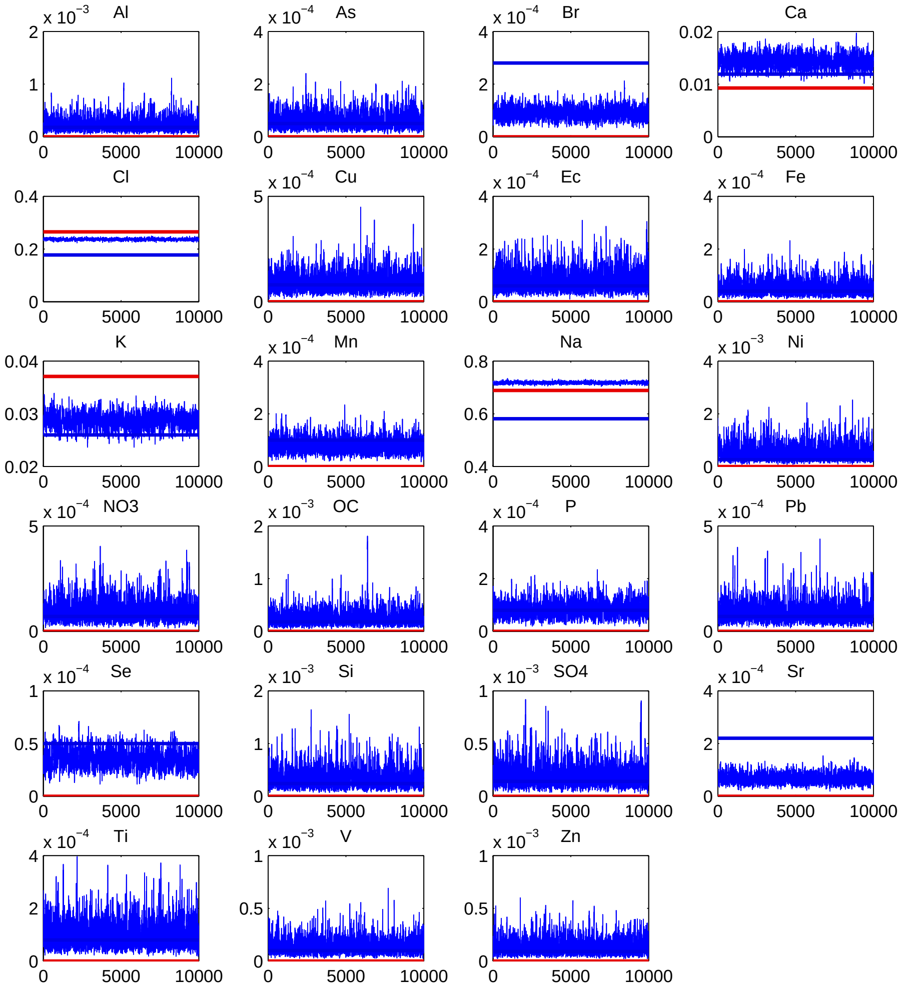


Figure 5.5: Trace plots for sea salt

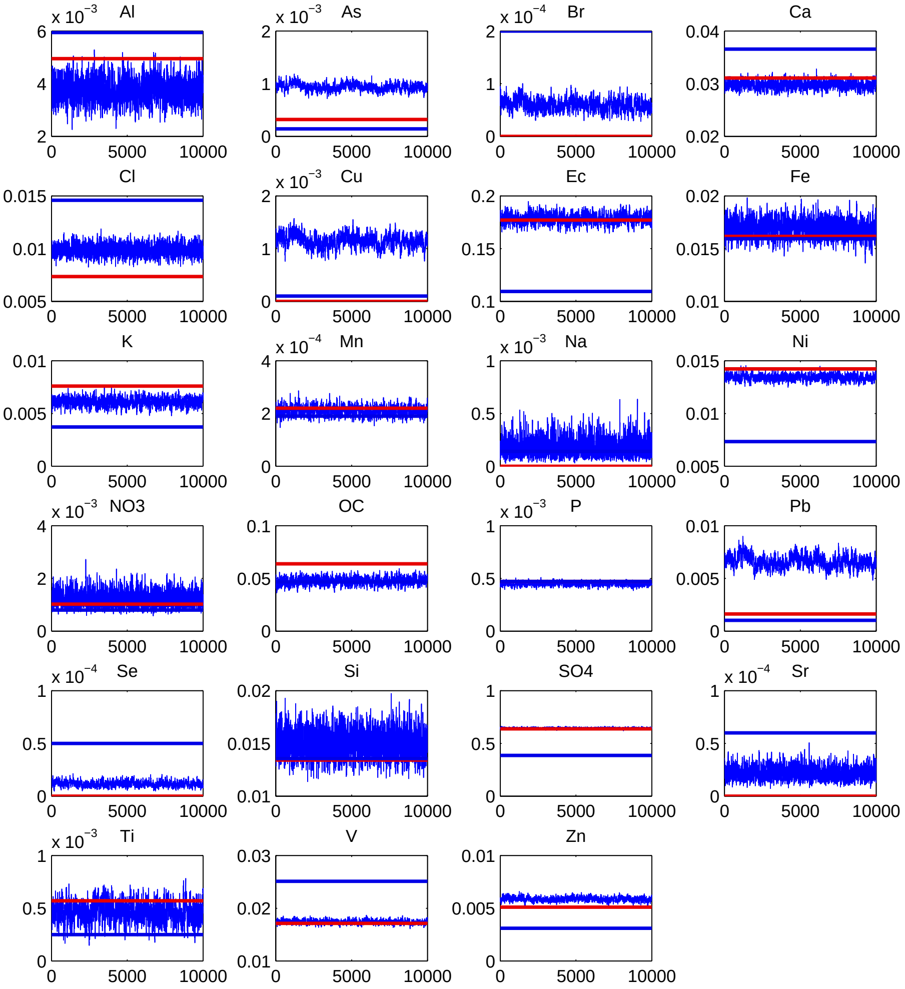


Figure 5.6: Trace plots for secondary sulfate

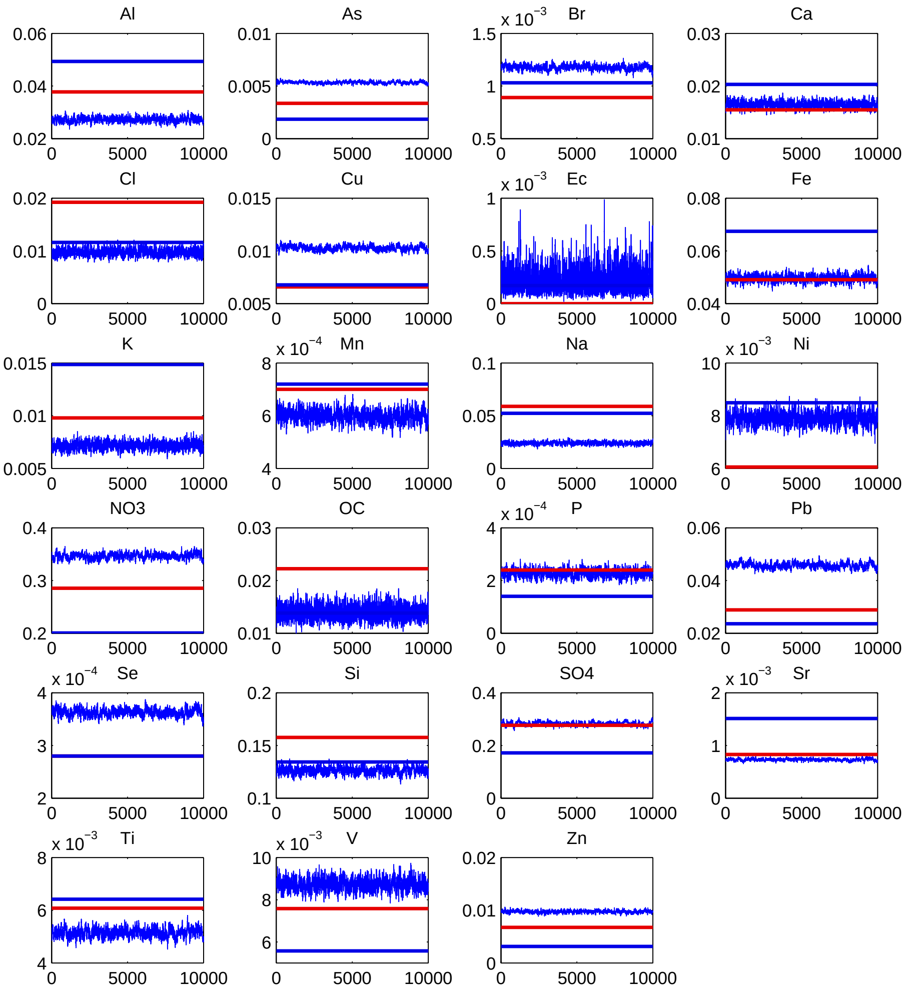


Figure 5.7: Trace plots for winter secondary

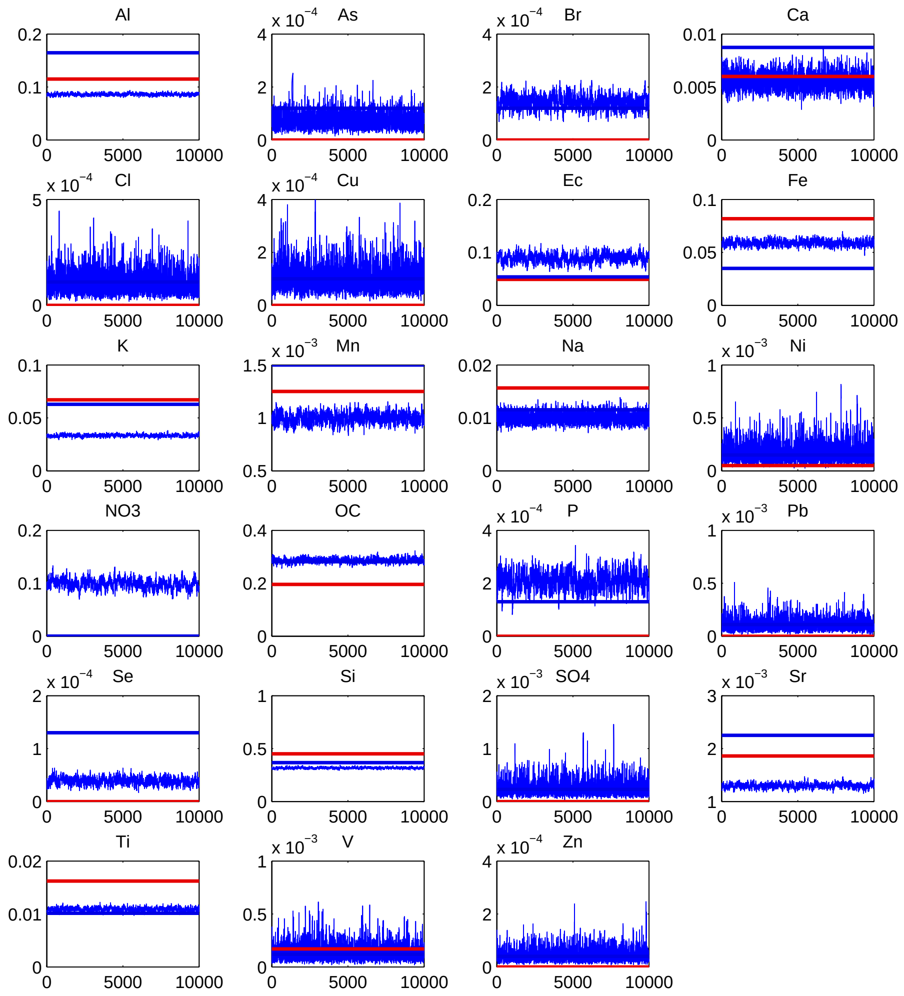


Figure 5.8: Trace plots for soil

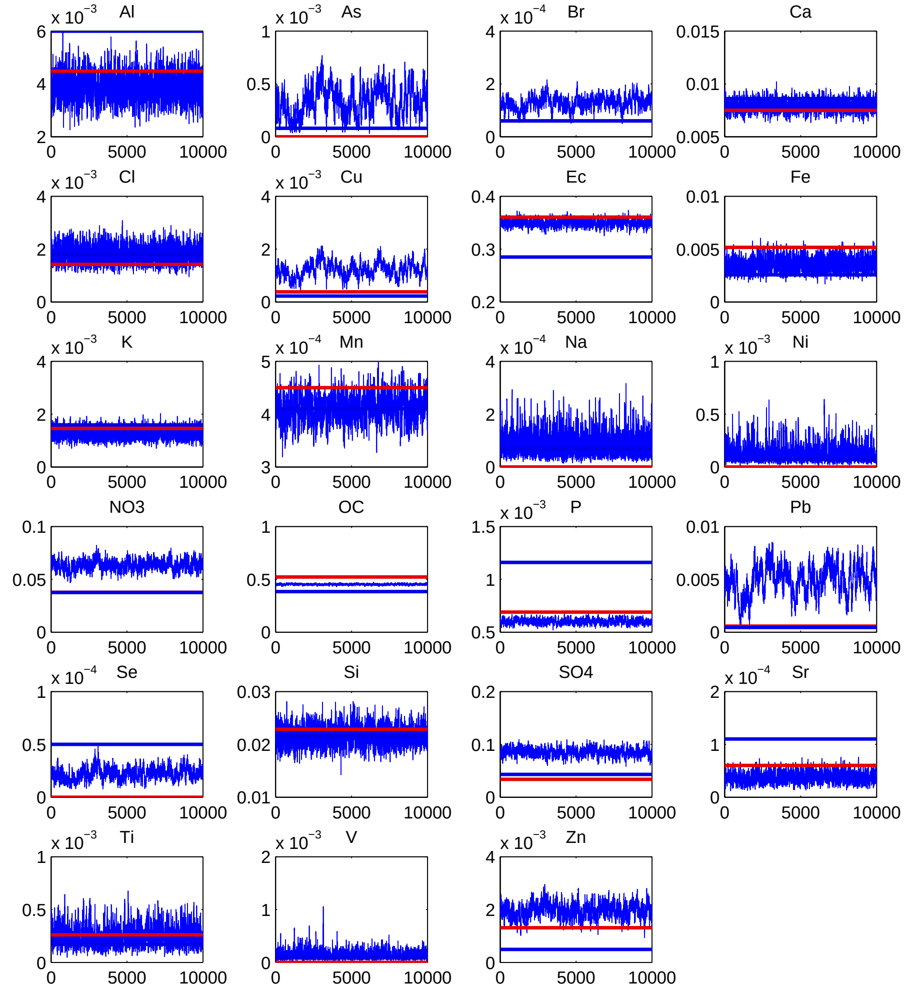


Figure 5.9: Trace plots for auto-diesel

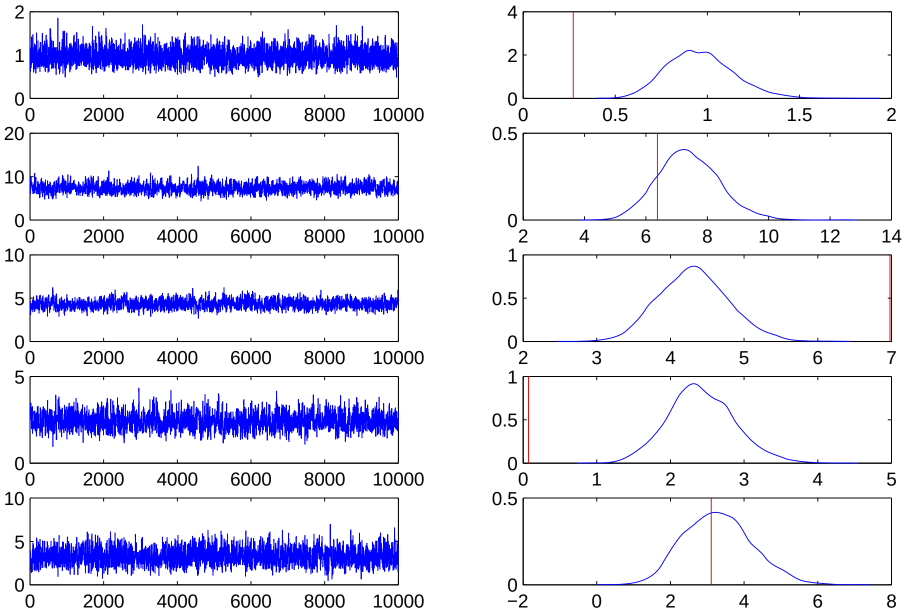


Figure 5.10: Trace plots and density estimates for the elements of $\hat{\mathbf{F}}$ on day 200.

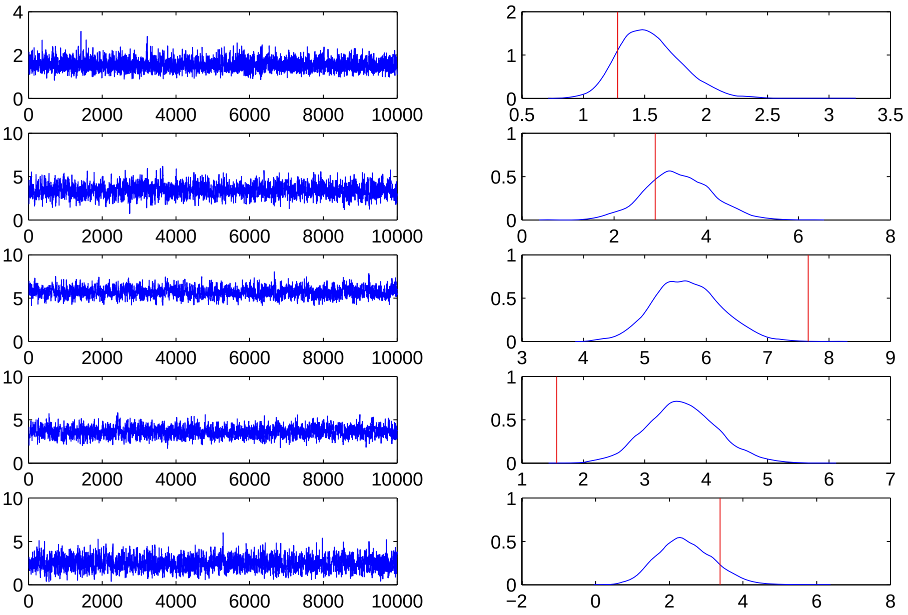


Figure 5.11: Trace plots and density estimates for the elements of $\hat{\mathbf{F}}$ on day 400.

For PMF without using any *a priori* information, the AAE for $\hat{\mathbf{F}}$ is 4.34, better than the results for the Bayesian method, and the AAE for $\hat{\mathbf{\Lambda}}$ is 0.127. For PMF using *a priori* information, with the prior means given as the target profiles and the prior variances given as the uncertainties, the AAE for $\hat{\mathbf{F}}$ is 5.521, and the AAE for $\hat{\mathbf{\Lambda}}$ is 0.122. The third *lims* value was changed from 0.003 to 0.3 so that PMF would converge.

The Effective Variance (EV) solution, another standard in the field that assumes knowledge of source profiles with some uncertainty (see Christensen *et al.*, 2006), gives estimation of $\mathbf{F}$ but not the source profiles. The AAE for the EV solution is 4.219.

A more suitable metric than the AAE for these Bayesian methods might be Total Median Absolute Error (TMAE), obtained in the following manner. Let $\hat{\mathbf{\Lambda}}$ be obtained from the posterior medians, instead of the posterior means. Let TMAE be the sum of the median absolute error of the $k$ sources in the model, where the median absolute error for the $h$th source is calculated as the median of $|\hat{\lambda}_{ih} - \lambda_{ih}|$ over $i$ for the $h$th source.

The TMAE for $\hat{\mathbf{\Lambda}}$ obtained as the posterior medians is 0.00348, while the TMAE for the means of the prior distributions on $\mathbf{\Lambda}$ is 0.00419. The TMAE for $\hat{\mathbf{F}}$ obtained as the posterior medians is 5.5202. In comparison, the TMAE for PMF without *a priori* information is 0.03257 for $\hat{\mathbf{\Lambda}}$ and 3.2407 for $\hat{\mathbf{F}}$. The TMAE for PMF using target profiles is 0.027563 for $\hat{\mathbf{\Lambda}}$ and 4.15999 for $\hat{\mathbf{F}}$. The TMAE for the EV solution is 2.95159.

Our Bayesian algorithm performs comparably to PMF and the EV solution. PMF and the EV solution estimate $\mathbf{F}$ better, but our algorithm does better than PMF in estimating $\mathbf{\Lambda}$, even when PMF uses target profiles. It is possible that changing uncertainties or settings of PMF could bring different results, but these at least give a rough idea of PMF's performance in comparison.

# Appendix A

# R code for studying PMF

## A.1 Generating data

```
setwd('c:/work/pmf_research')


prof<-as.matrix(read.table("trueprofiles.txt"))

names(prof) <- c('salt','sec_sulf','sec_1','soil','autod')

fac<-as.matrix(read.table("truesources.txt"))


fprof <- (read.table("wfcnewsourceprofiles.csv",sep=",",header=T))

ffac <- (read.table("wfcnewsourcecontributions.csv",sep=",",header=T))


fullprof <- as.matrix(fprof[,-1])

fullfac <- as.matrix(ffac[,-1])




srcnums <- c(7,2,3,4,6)

newwts <- diag(c(.2,.5,.5))


CCV <- .5
```

```
truedat <- t(prof %*% t(fac))


gendat(.5,"c:/work/pmf_research/true_.5/",F)

gendat(.5,"c:/work/pmf_research/miss_.5/",T)

gendat(1,"c:/work/pmf_research/true_1/",F)

gendat(1,"c:/work/pmf_research/miss_1/",T)


uncertainties <- round( fullprof[,srcnums] %*% t(fullfac[,srcnums]) * .5 , 8)

write(uncertainties, file="miss_.5/uncertainties.txt", ncol=nrow(uncertainties))


uncertainties <- round( fullprof[,srcnums] %*% t(fullfac[,srcnums]) * .5 , 8)

write(uncertainties, file="true_.5/uncertainties.txt", ncol=nrow(uncertainties))


uncertainties <- round( fullprof[,srcnums] %*% t(fullfac[,srcnums]) * 1 , 8)

write(uncertainties, file="miss_1/uncertainties.txt", ncol=nrow(uncertainties))


uncertainties <- round( fullprof[,srcnums] %*% t(fullfac[,srcnums]) * 1 , 8)

write(uncertainties, file="true_1/uncertainties.txt", ncol=nrow(uncertainties))


mywrite.table <- function(x,path){

write.table(x,path,col.names=F,row.names=F,quote=F)

}


gendat <- function(CCV,path,plussource){

fprof <- (read.table("wfcnewsourceprofiles.csv",sep=",",header=T))

ffac <- (read.table("wfcnewsourcecontributions.csv",sep=",",header=T))
```

```
fullprof <- as.matrix(fprof[,-1])

fullfac <- as.matrix(ffac[,-1])

srcnums <- c(7,2,3,4,6)

newwts <- diag(c(.2,.5,.5))

truedat <- t(prof %*% t(fac))

n<-788;p<-23;

for(i in 1:100){

ydat <- array(dim=c(n,p))

if (plussource == F)

ydat <- exp(  log(truedat) - .5*log(CCV^2 +1) +

sqrt(log(CCV^2+1)) * matrix(rnorm(n*p),n,p)  )

if (plussource==T) {

ydat <- exp(  log(fullprof[,srcnums] %*% t(fullfac[,srcnums]) +

       fullprof[,-srcnums] %*% t(fullfac[,-srcnums] %*% newwts))

       - .5*log(CCV^2 +1) + sqrt(log(CCV^2+1)) * matrix(rnorm(p*n),p,n)  )

ydat <- t(ydat)}

mywrite.table(round(ydat,6),paste(path,'ydat',i,'.txt',sep=""))}}


###########################generate Ahat matrices

ACV <- .25

A <- fullprof[,srcnums]

Aplus <- A

Aplus[A < .0001] <- .0001

p <- 23

for(k in 1:100){

Ahat <- exp( log(Aplus) -.5*log(ACV^2 +1) +
               sqrt(log(ACV^2+1)) * ( matrix(rnorm(p*5),p,5) ) )
for (i in 1:5) if (sum(Ahat[,i]) > 1) Ahat[,i] <- Ahat[,i]/sum(Ahat[,i])
mywrite.table(round(Ahat,6),paste('Ahats/Ahat_25_',k,'.txt',sep=""))
}


ACV <- 1.0
for(k in 1:100){
Ahat <- exp( log(Aplus) -.5*log(ACV^2 +1) +
               sqrt(log(ACV^2+1)) * ( matrix(rnorm(p*5),p,5) ) )
for (i in 1:5) if (sum(Ahat[,i]) > 1) Ahat[,i] <- Ahat[,i]/sum(Ahat[,i])
mywrite.table(round(Ahat,6),paste('Ahats/Ahat_1_',k,'.txt',sep=""))
}
```

## A.2 Example of R documentation for package *pmf*

```
runpmf                  package:pmf                  R Documentation


Run PMF


Description:


     Runs PMF.  May use target profiles ("Gkey").  Target profiles are
     appended to the data matrix and their uncertainties appended to
     the matrix of ambient error measurements.  The 'Gkey' is the
     matrix of integer values used by PMF to pull these to zero (see
     the PMF documentation).  If target profiles are used, the .ini
     file must be set up to do so.
```

Usage:

```
     runpmf(inifilename, data, F, Lambda, Q, sources = 5,
     rows = 788, cols = 23, otd = 4, fpeak = 0, c1 = 0,
      c2 = 0, c3 = 0.01, em = 12, startval = TRUE, gnot,
     fnot, stdevs = "uncertainties.txt", fkey = FALSE, fkeyname)
```

Arguments:

```
inifilename: Name of the .ini file

       data: Name of the data file

          F: Name of output file for F

     Lambda: Name of output file for Lambda

          Q: Name of output file for Q

    sources: Number of sources

       rows: Number of rows in data matrix

       cols: Number of columns in data matrix
     otd: Outlier Threshold Distance

   fpeak: Fpeak

      c1: C1

      c2: C2

      c3: C3

      em: Errormodel

startval: Logical: use starting values?

    gnot: Starting values for F

    fnot: Starting values for Lambda

  stdevs: Uncertainties of the data

    fkey: Logical: use fkey? (Pull elements to zero)

fkeyname: If an fkey is used, name of the file

Details:
```

The .ini file should have the gkey read in as file option 4, with the Gkey appended. This function modifies some BUT NOT ALL of the PMF settings. Other settings can be changed by modifying the .ini file. Some settings may need to be changed for PMF to converge using target profiles.

Note:

Make sure starting values are appropriately sorted! .ini filename cannot contain periods. Note that PMF will stick an extra k rows on top of the output F matrix. These rows need to be deleted before analysis takes place!

Author(s):

Jeff Lingwall

References:

See PMF documentation

Examples:

```
#you need pmf2wtst.exe in the current directory
##Example without target profiles
data(default_ini)
```

```
write(default,file="default.ini") #writes

 the .ini file in the current directory

data(uncertainties)

cwrite.table(uncertainties,"uncertainties.txt")

data(ydat)

cwrite.table(ydat,"ydat.txt")

runpmf("default.ini", "ydat.txt", "myF.txt", "myLambda.txt",

"myQ.txt", sources = 5,

rows = 788, cols = 23, startval = FALSE, stdevs = "uncertainties.txt")

data(truesources)

     data(trueprofiles)

     sort <- msesort(t(read.table("myLambda.txt")),

     read.table("myF.txt"),truesources)

     AAE(sort$fac,truesources)

     AAE(sort$lam,trueprofiles)


##Target profile example

data(gkeyini)

     write(gkeyinifile,file="myinifile.ini") #writes the

      .ini file in the current directory

     data(uncertainties)

     cwrite.table(uncertainties,"uncertainties.txt")

     data(ydat)

     cwrite.table(ydat,"ydat.txt")

     data(Ahat)

     tfprep(Ahat,Ahat*.2,p=23,k=5,"ydat.txt",keyname="keydat.txt",
     uncname="uncertainties.txt",

     keyuncname="keyuncertainties.txt")

     runpmf("myinifile.ini", "keydat.txt", "myF.txt",

     "myLambda.txt", "myQ.txt",

     sources = 5, rows = 788+5, cols = 23, startval = FALSE,

     stdevs = "keyuncertainties.txt")

     data(truesources)

     data(trueprofiles)

     sort <- msesort(t(read.table("myLambda.txt")),

     read.table("myF.txt")[-c(1:5),],

     truesources)

     AAE(sort$fac,truesources)

     AAE(sort$lam,trueprofiles)
```

# Appendix B

## MATLAB code to fit a Bayesian pollution source apportionment model

```
%%%%%%%%%%%%%%%%%
%
%         Jeff Lingwall
%
%         Bayesian pollution source apportionment
%
%
%%%%%%%%%%%%%%%%%
%% Define things
% global variables
global d t tau delta length burn days y k ksi
mutrans_constant; %make prior info global variables

%  data
ydat = transpose(dlmread('ydat1.txt'));
tlam = dlmread('trueprofiles.txt');
tfac = dlmread('truesources.txt');
lam = dlmread('Ahat.txt');
```

```
scale = inline(’x./sum(x)’);

for t = 1:5
    lam(:,t) = scale(lam(:,t));
end

lam_unc = lam;

ind = find(lam >=.4);
lam_unc(ind) = .02 * lam_unc(ind);

ind = find(lam < .4 & lam >= .1);
lam_unc(ind) = .05 * lam_unc(ind);

ind = find(lam < .1 & lam >=.01);
lam_unc(ind) = .1 * lam_unc(ind);

ind = find(lam < .01 & lam >=.001);
lam_unc(ind) = .2 * lam_unc(ind);

ind = find(lam < .001);
lam_unc(ind) = .5 * lam_unc(ind);

d = log(lam) - .5 .* log( (lam .^ 2 + lam_unc .^2)./lam .^2 ) ;
t = log( (lam .^ 2 + lam_unc .^2) ./ lam .^2) ;
k = .2;
```

```
ksi =  sqrt(log( k^2 + 1));
mutrans_constant =  .5 * log( k^2 + 1 );

% prior on f  (mean and variance of the log of the data)
tau = 2;
delta = 1;

% mcmc settings
days = 788;
length = 10000;
burn =0;

%% starting points, data
f = dlmread('starting_points_f.txt');
candsig_f = dlmread('candsigs_for_f.txt');
l = dlmread('starting_points_l.txt');% lam;
candsig_l =dlmread('candsigs_for_l.txt');
% f = ones(5,days) .* 10;
% candsig_f = ones(5,days);
% l = lam;
% candsig_l = .2 * lam;

%acceptance ratios, etc
l_rat = zeros(23,5);
f_rat = zeros(5,days);
y = (ydat(:,1:days));
```

```
%% print out values for short run
dlmwrite('f.txt',f);
dlmwrite('l.txt',l);
%% print out values for long run
% dlmwrite('f_sample.txt',f);
% dlmwrite('l_sample.txt',l);

%% mcmc
count = 1;
intval = 20; %every intval iterations, adjust the candidate
sigmas and print out stuff
% tic
for i = 2:(length+burn)
    if mod(i,intval) == 0
        i
    end

    accept_f = zeros(5,days);
    count = count+1;
    for ii = 1:5
        old = f(ii,:);
        new = normrnd(old,candsig_f(ii,:));
        zero_index = find(new <= 0);
        new(find(new <=0)) = old(find(new<=0));
        f_update = f;
        f_update(ii,:) = new;
        p_F = (log(lognpdf(f(ii,:),tau,sqrt(delta))));
        p_F_update = (log(lognpdf(new,tau,sqrt(delta))));
        llo = posterior_f(f,y,l,p_F);
        lln = posterior_f(f_update,y,l,p_F_update);
        uu = rand(788,1)';
        ind = find(log(uu) < (lln - llo));
        f(ii,ind) = new(ind);
        accept_f(ii,ind) = 1;
        accept_f(ii,zero_index) = 0;      %these didn't really change
    end

    accept_l = zeros(23,5);
    for kk = 1:5
        for k = 1:23
            old = l(k,kk);
            l_update = l;
            new = normrnd(old,candsig_l(k,kk));
            if new > 0 & new <=1
                l_update(k,kk) = new;
                l_update(:,kk) = scale(l_update(:,kk));
                p_lam = sum(sum(log(mylognpdf(l,d,sqrt(t)))));
                p_lam_update = sum(sum(log(mylognpdf(l_update,d,sqrt(t)))));
                llo = posterior(f,y,l,p_lam);
                lln = posterior(f,y,l_update,p_lam_update);
                uu = rand(1);
                if log(uu) < (lln - llo);
                    l = l_update;
                    accept_l(k,kk) = 1;
                end%
            end
        end
    end

    l_rat = (l_rat*(count-1) + accept_l) / count;
    f_rat = (f_rat*(count-1) + accept_f) / count;
    %adjust candidate sigmas

     if  i == intval
        dlmwrite('l_rat.txt',l_rat);
        dlmwrite('f_rat.txt',f_rat);
        dlmwrite('candsigs_for_l.txt',candsig_l);
        dlmwrite('candsigs_for_f.txt',candsig_f);
    end

     if mod(i,intval*10) == 0
        dlmwrite('l_rat.txt',l_rat);
        dlmwrite('f_rat.txt',f_rat);
        dlmwrite('candsigs_for_l.txt',candsig_l);
        dlmwrite('candsigs_for_f.txt',candsig_f);
    end
```

```
    if i <= burn & mod(i,intval) == 0
        candsig_l(find(l_rat < .2)) =  candsig_l(find(l_rat < .2))* .9;
        candsig_l(find(l_rat > .6)) =  candsig_l(find(l_rat > .6))* 1.1;
        l_rat = zeros(23,5);
        candsig_f(find(f_rat < .25)) =  candsig_f(find(f_rat < .25))* .9;
        candsig_f(find(f_rat > .55)) =  candsig_f(find(f_rat > .55))* 1.1;
        f_rat = zeros(5,days);
        count = 1;
    end
    %%%%%%%%%%%%%%%%%%%%% write out values in a short run
    dlmwrite('f.txt',f,'-append');
    dlmwrite('l.txt',l,'-append');
        %%%%%%%%%%%%%%%%%%%%%% write out values in a run of 100,000
%     if mod(i,intval) == 0
%         dlmwrite('f_sample.txt',f,'-append');
%         dlmwrite('l_sample.txt',l,'-append');
%     end
%     if i == 90000
%         dlmwrite('f.txt',f);
%         dlmwrite('l.txt',l);
%     end
%     if i > 90000
%         dlmwrite('f.txt',f,'-append');
%         dlmwrite('l.txt',l,'-append');
%     end
    %%%%%%%%%%%%%%%%%%%%%%%
```

```
end %end of i loop 1:(burn + length)


%% write files

dlmwrite('l_rat.txt',l_rat);

dlmwrite('f_rat.txt',f_rat);
```

# Appendix C

## MATLAB code for making three-dimensional movies

```
%% three d metropolis steps for day 100 of F

input = dlmread('f_sample.txt');

cols=  jet(24)

input(50001:50005,:) = [];

f = ones(5,788,2000);

count = 0;

for k = 0:5:(10000-1)

    count = count + 1;

    f(:,:,count) = input((1+5*k):(5+5*k),:);

end

clear input

burn = 1800j

testval = 100

count = 0

for j = 5:5:1800

    count = count + 1;


plot3(converter(f(1,testval,1:j)),converter(f(2,testval,1:j)),
converter(f(3,testval,1:j)),'Color',cols(1,:))
xlabel('F[1,100]');ylabel('F[2,100]');zlabel('F[3,100]')
        grid on
%         hold on
F(count) = getframe(gcf);
end
for j = (1800+5):5:(2000 - 5)
    count = count + 1;
        plot3(converter(f(1,testval,1:burn)),converter(f(2,testval,1:burn)),
        converter(f(3,testval,1:burn)),'Color',cols(1,:))
        xlabel('F[1,100]');ylabel('F[2,100]');zlabel('F[3,100]')
        hold on
        plot3(converter(f(1,testval,(burn):j)),converter(f(2,testval,(burn):j)),
        converter(f(3,testval,(burn):j)),'Color',cols(10,:))
        xlabel('F[1,100]');ylabel('F[2,100]');zlabel('F[3,100]')
        grid on
        hold off
        F(count) = getframe(gcf);
end
F(1) = F(399)
movie2avi(F,'test.avi')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% rendered surfaces
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%write out code to R
```

```
% kde2d in R, write back to MATLAB, one for each day

for i = 1:788
dlmwrite(strcat('f1_',int2str(i)),f(1,i,:))
dlmwrite(strcat('f2_',int2str(i)),f(2,i,:))
end

%% looop through plotting each day (factors 1 and two) !!!! with names!!!
clear F;
counter = 0;
counter2 = 0;
colorcount = 1;
ctrip = 1;
cols = jet(100);
for i = 1:200
x = dlmread(strcat('Rf1_',int2str(i)));
y = dlmread(strcat('Rf2_',int2str(i)));
z = dlmread(strcat('RZ_',int2str(i)));
surf(x,y,z,'FaceColor',cols(colorcount,:),'EdgeColor','none')
junk = axis;
 axis([0 10 0 20 junk(5) junk(6)]);
if i == 1
[az,el] = view;
end
```

```
if ctrip == 1

colorcount = colorcount + 1;

end

if ctrip == 0

colorcount = colorcount - 1;

end

if colorcount == 100

    ctrip = 0;

end

if colorcount == 1

    ctrip = 1

end




counter = counter + 1;

if el + counter2 > 70

    trip = 0;

end

if el + counter2 < 46

    trip = 1;

end

if trip == 1

    counter2 = counter2 + 2;

end

if trip == 0

    counter2 = counter2 - 2;
```

```
end

view(az + counter,el + counter2);

xlabel('Sea salt');ylabel('Secondary sulfate');zlabel(int2str(i));

camlight left; lighting phong

F(i) = getframe(gcf);

end

%%

movie2avi(F,'render.avi','fps',3,'quality',100)%,'fps',1
```

# Bibliography

Bayes, T. (1763), "An essay towards solving a problem in the doctrine of chances," *Philosophical Transactions of the Royal Society of London*, 53, 370–418, cited in Dale (1999).

Billera, L. J. and Diaconis, P. (2001), "A Geometric Interpretation of the Metropolis-Hastings Algorithm," *Statistical Science*, 16, 335–339.

Blifford, I. and Meeker, G. (1967), "A factor analysis model of large scale pollution." *Atmospheric Environment*, 1, 147–154.

Christensen, W. F. and Gunst, R. F. (2004), "Measurement error models in chemical mass balance analysis of air quality data," *Atmospheric Environment*, 38, 733–744.

Christensen, W. F. and Sain, S. R. (2002), "Accounting for dependence in a flexible multivariate receptor model," *Technometrics*, 44, 328–337.

Christensen, W. F., Schauer, J. J., and Lingwall, J. W. (2006), "Iterated Confirmatory Factor Analysis for Pollution Source Apportionment," *Environmetrics.*

Dale, A. I. (1999), *A History of Inverse Probability: From Thomas Bayes to Karl Pearson*, New York: Springer-Verlag, 2nd ed.

Dockery, D. W., Pope, C. A., Xu, X., Spengler, J. D., Ware, J. H., Fay, M. E., Ferris, B. G., and Speizer, F. E. (1993), "An Association between Air Pollution and Mortality in Six U.S. Cities," *The New England Journal of Medicine*, 329, 1753–1759.

Dominici, F., Samet, J. M., and Zeger, S. L. (2000), "Combining Evidence on Air Pollution and Daily Mortality from the 20 Largest US Cities: A Hierarchical Modelling Strategy," *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 163, 263–302.

Eberly, S. (2005), "EPA PMF 1.1 User's Guide," *U.S. Environmental Protection Agency.*

Eppstein, M. J., Hawrysz, D. J., Godavarty, A., and Sevick-Muraca, E. M. (2002), "Three-Dimensional, Bayesian Image Reconstruction from Sparse and Noisy Data Sets: Near-Infrared Fluorescence Tomography," *Proceedings of the National Academy of Sciences of the United States of America*, 99, 9619–9624.

Ferraz, G., Russell, G. J., Stouffer, P. C., Richard O. Bierregaard, J., Pimm, S. L., and Lovejoy, T. E. (2003), "Rates of Species Loss from Amazonian Forest Fragments," *Proceedings of the National Academy of Sciences of the United States of America*, 100, 14069–14073.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian Data Analysis*, Washington, D.C.: Chapman and Hall / CRC, 2nd ed.

Geweke, J., Gowrisankaran, G., and Town, R. J. (2003), "Bayesian Inference for Hospital Quality in a Selection Model," *Econometrica*, 71, 1215–1238.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996), "Introducing Markov chain Monte Carlo," in *Markov Chain Monte Carlo in Practice*, eds. Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., Chapman and Hall/CRC, pp. 1–19.

Grozinger, C. M., Sharabash, N. M., Whitfield, C. W., and Robinson, G. E. (2003), "Pheromone-Mediated Gene Expression in the Honey Bee Brain," *Proceedings of the National Academy of Sciences of the United States of America*, 100, 14519–14525, arthur M. Sackler Colloquium on Chemical Communication in a Post-Genomic World.

Henry, R. C. (1991), "Multivariate receptor models," in *Receptor modeling for air quality management*, ed. Hopke, P. K., Elsevier, vol. 7 of *Data handling in science and technology*, pp. 117–147.

— (2003), "Multivariate receptor modeling by N-dimensional edge detection." *Chemometrics & Intelligent Laboratory Systems*, 65, 179–189.

Hopke, P., Gladney, E., Gordon, G., Zoller, W., and Jones, A. (1976), "The Use of Multivariate Analysis to Identify Sources of Selected Elements in the Boston Urban Aerosol," *Atmospheric Environment*, 10, 1015–1025.

Hopke, P. K., Kim, E., Larson, T. V., and Covert, D. S. (2004), "Analysis of Ambient Particle Size Distributions Using Unmix and Positive Matrix Factorization," *Environmental Science and Technology*, 38, 202–209.

Hopke, P. K., Xie, Y., and Paatero, P. (1999), "Mixed multiway analysis of aiborne particle composition data," *Journal of Chemometrics*, 13, 343–352.

Javitz, H., Robinson, N., and Watson, J. (1988), "Performance of the chemical mass balance model with simulated local-scale aerosols." *Atmospheric Environment*, 22, 2309–2322.

Lee, E., Chan, C. K., and Paatero, P. (1999), “Application of positive matrix factorization in source apportionment of particulate pollutants in Hong Kong,” *Atmospheric Environment*, 33, 3201–3212.

Miller, M., Friedlander, S., and Hidy, G. (1972), “A chemical element balance for the Pasadena aerosol.” *Journal of Colloid and Interface Science*, 39, 165–176.

Miller, R. E., Buckley, T. R., and Manos, P. S. (2002), “Examination of the Monophyly of Morning Glory Taxa Using Bayesian Phylogenetic Inference,” *Systematic Biology*, 51, 740–753.

Paatero, P. (1997), “Least squares formulation of robust non-negative factor analysis,” *Chemometrics and Intelligent Laboratory Systems*, 37, 23–35.

— (2004a), *User's Guide for Positive Matrix Factorization programs PMF2 and PMF3, Part 1: tutorial.*

— (2004b), *User's Guide for Positive Matrix Factorization programs PMF2 and PMF3, Part 2: reference.*

Paatero, P., Hopke, P. K., Hoppenstock, J., and Eberly, S. I. (2003), “Advanced Factor Analysis of Spatial Distributions of PM2.5 in the Eastern United States,” *Environmental Science & Technology*, 37, 2460–2476.

Paatero, P. and Tapper, U. (1994), “Positive Matrix Factorization: a non-negative factor model with optimal utilization of error estimates of data values,” *Environmetrics*, 5, 111–126.

Park, E. S., Guttorp, P., and Henry, R. C. (2001), “Multivariate Receptor Modeling for Temporally Correlated Data by Using MCMC,” *Journal of the American Statistical Association*, 96, 1171–1183.

Thomas, D. C. (2000), “Some Contributions of Statistics to Environmental Epidemiology,” *Journal of the American Statistical Association*, 95, 315–319.