ELSEVIER

Chemometrics and Intelligent Laboratory Systems 37 (1997) 23–35

Chemometrics and intelligent laboratory systems

Least squares formulation of robust non-negative factor analysis

Pentti Paatero

Department of Physics, University of Helsinki, Box 9, FIN-00014 Helsinki, Finland

Abstract

Positive matrix factorization (PMF) is a recently published factor analytic technique where the left and right factor matrices (corresponding to scores and loadings) are constrained to non-negative values. The PMF model is a weighted least squares fit, weights based on the known standard deviations of the elements of the data matrix. The following aspects of PMF are discussed in this work: (1) Robust factorization (based on the Huber influence function) is achieved by iterative reweighting of individual data values. This appears especially useful if individual data values may be in error. (2) Desired rotations may be obtained automatically with the help of suitably chosen regularization terms. (3) The algorithms for PMF are discussed. A synthetic spectroscopic example is shown, demonstrating both the robust processing and the automatic rotations.

Keywords: Positive matrix factorization; Iterative reweighting; Huber influence function

1. Introduction

Traditionally the factor analytic problem has been approached from the viewpoint of correlations. This approach stems originally from the soft sciences where the modelling is usually qualitative. In physical sciences such as spectroscopy and chemometrics there are often underlying physical models possessing the property of linear additivity. An example is Beers' Law for the additivity of absorbances of the components of a mixture. Also, the data are precise and there are not many systematic errors. Then a quantitative approach is needed. We are accustomed to discuss data fitting and we wish to fit up to the full precision of the data.

The traditional PCA is equivalent to a least squares (LS) fit to the matrix. However, it was demonstrated by Paatero and Tapper [1] that this LS fit is badly weighted, implicitly assuming non-realistic standard deviations for the values of the data matrix. Thus the results of PCA cannot be of minimum variance as they are not based on a correct weighting.

A different approach to factor analysis has been published by Paatero and Tapper [2]. The model is set up as a weighted least squares task. It is assumed that a standard deviation $\boldsymbol{\sigma}_{ij}$ is known for each data value $\mathbf{X}_{ij}$. Thus optimum LS weights for each value $\mathbf{X}_{ij}$ are $\mathbf{w}_{ij} = 1/\boldsymbol{\sigma}_{ij}^2$. There are important connections between the physical reality and the values $\boldsymbol{\sigma}_{ij}$, especially in environmental models. These connections have been discussed in that paper and this discussion is not repeated. In many physical situations negative values of the factors are not meaningful as such. For this reason non-negativity has been included in the model. The results have been quantitative. In precipitation studies it was possible to compute anion–cation balances of the *composition factors*, see Juntto and Paatero [3], Anttila et al. [4].

This paper discusses some unpublished aspects of the new model: An efficient algorithm PMF2 is in-

PII S0169-7439(96)00044-5

troduced. It converges significantly faster than the previously used ALS (alternating least squares). It is also shown how iterative reweighting allows one to implement a robust analysis which is more tolerant of outliers than the customary techniques of factor analysis. It is demonstrated how the inclusion of special regularization terms in the enhanced object function enables one to obtain differently rotated solutions.

A synthetic spectroscopy-like example demonstrates these aspects.

2. Positive matrix factorization PMF

2.1. The model

The symbols **G** and **F** denote the left and right factor matrices to be determined. They correspond to the scores and loadings. Define the 'residual matrix' **E**, the difference between measurement $\mathbf{X}_{ij}$ and model $\mathbf{Y}_{ij}$, as a function of the factors **G** and **F**, as

$$\mathbf{E}_{ij} = \mathbf{X}_{ij} - \mathbf{Y}_{ij} = \mathbf{X}_{ij} - \sum_{h=1}^{p} \mathbf{G}_{ih}\mathbf{F}_{hj} \quad (i = 1, \dots, m,\ j = 1, \dots, n). \tag{1}$$

Define the 'object function' Q, to be minimized, as a function of the factors **G** and **F**, as

$$Q(\mathbf{E}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(\mathbf{E}_{ij}/\boldsymbol{\sigma}_{ij}\right)^2. \tag{2}$$

The values $\boldsymbol{\sigma}_{ij}$ are the standard deviations of the observed values $\mathbf{X}_{ij}$. The task of the non-negatively constrained weighted factor analysis is: Minimize $Q(\mathbf{E})$ with respect to **G** and **F** under the constraint that all or some of the elements of **G** and **F** are constrained to non-negative values. We call this task positive matrix factorization (PMF), with the acronyms PMF2 and PMF3 denoting the programs used in two and three dimensions.

2.2. Extending the model to three dimensions

The 2-dimensional factor analytic models may be extended to three dimensions in different ways. We only consider the PARAFAC model which is defined by

$$\mathbf{E}_{ijk} = \mathbf{X}_{ijk} - \sum_{h=1}^{p} \mathbf{A}_{ih}\mathbf{B}_{jh}\mathbf{C}_{kh} \quad (i = 1, \dots, m,\ j = 1, \dots, n,\ k = 1, \dots, o). \tag{3}$$

Here **X** is the observed 3-way data array and **E** the corresponding array of residual values. The matrices **A**, **B**, and **C** are the 'factor matrices', sometimes called 'modes'. In the original definition of PARAFAC by Harshman and Lundy [5] the object function

$$Q(\mathbf{E}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{o} \mathbf{E}_{ijk}^2 \tag{4}$$

was minimized without regard to non-negativity. Now we define the function Q analogously to Eq. (2) as

$$Q(\mathbf{E}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{o} \left(\mathbf{E}_{ijk}/\boldsymbol{\sigma}_{ijk}\right)^2. \tag{5}$$

The object function Q is minimized under the constraint that all elements of the three factor matrices **A**, **B**, and **C** must be non-negative. This model has originally been solved by Ross and Leurgans [6].

The present paper mostly discusses work related to the two-dimensional variant of the model PMF. However, most of the results (excluding the techniques for producing desired rotations) have also been extended to three dimensions. Papers describing the 3-way results are in preparation.

2.3. Tradeoffs of the PMF approach

The possibility of solving the factorization problem by Eigenanalysis should be considered as a 'special case'. It is a 'mathematical coincidence' that such an elegant possibility exists at all. Weighting the LS fit prevents using Eigenanalysis except when the matrix of weights is of rank = 1. Doing 3-way (PARAFAC) analysis also prevents straightforward use of Eigenanalysis. In these cases the least squares approach is the only possibility.

Constraining the factors to non-negative values may also be quite complicated with Eigenanalysis.

The straightforward approach would be to take the p most significant pairs of singular vectors and to rotate them until no negative values remain. This should be always possible if non-negativity is only required for the left factor matrix **G** ('scores') or only for the right factor matrix **F** ('loadings'). It was pointed out by Paatero and Tapper [2] that with some matrices the rotation is unable to produce strictly non-negative values for both the left and the right factors simultaneously. In such cases the solution may only be found so that the Eigensolution is modified with some kind of iterative process.

The tradeoffs have been examined in more detail by Paatero and Tapper [2], the following is a summary of them.

2.4. Properties of the Eigenanalysis

The computation of an Eigenanalysis is fast. Another good property is that the result is unique in the sense that there is only one minimum of the object function Q, there are no local minima.

A drawback is that the unique minimum value of Q is achieved by infinitely many different solutions, i.e. infinitely many 'rotations' are possible. In many cases (especially in physical and environmental sciences) the negative values of factors have no quantitative interpretation.

2.5. Properties of the weighted constrained least squares

The weighted constrained least squares fit has a number of desirable properties: depending on the data, sometimes there is no rotational ambiguity at all. Non-negative factors are more meaningful in many applications, they have a quantitative interpretation whereas negative values would only have a qualitative meaning. Because of the weighting, information is utilized optimally, the results have the property of minimum variance. Especially in environmental research, special application-oriented information may be conveyed to the model by means of the matrix $\boldsymbol{\sigma}_{ij}$ of standard deviations of data values $\mathbf{X}_{ij}$.

The main weakness of weighted constrained least squares is that the object function Q may possess several local minima, leading to multiple solutions of the factorization problem. Also, the computation may be slow for large problems.

2.6. Multiple solutions

The presence of multiple solutions may be investigated so that one reruns the algorithm using several different sets of pseudorandom values as the initial points. If the problem at hand has local solutions then different runs will result in different values for the matrices **G** and **F** and also for Q. There remains the problem of selecting among these different results. It has been suggested to us that the different Q values could be used to construct 'probabilities' for different solutions. This is an open research problem.

Our current advice is that one should regard all the solutions more or less independently from each other. Inspect all of them and either accept the most sensible of them if a meaningful choice is possible or else regard the solution as ambiguous, perhaps reporting two different explanations for the same data.

The problem of multiple solutions seems not to be very essential in practice. Our informal experience suggests that multiple optima mostly tend to occur in certain special circumstances. Firstly, if there are p factors in the matrix but the analysis is performed with $p-1$ factors, different solutions may leave different factors unexplained. On the other hand, if there are p factors in the matrix but the analysis is performed with $p+1$ factors, the 'extra' factor is able to explain some of the noise in the matrix and different solutions may explain different parts of the noise. Finally, if the pth singular value of the matrix is very small, almost down to noise level, there may be a solution which 'correctly' explains those p factors but also solutions which explain $p-1$ factors and some noise. In summary, multiple solutions seem to occur mostly when the model is in some way questionable.

2.7. The enhanced object function Q

When a non-negatively constrained problem is solved in practice the function to be minimized at each step of the iteration is the following enhanced $\bar{Q}$.

$$\begin{aligned}
&\bar{Q}(\mathbf{E}, \mathbf{G}, \mathbf{F}) \\
&\quad = Q(\mathbf{E}) + P(\mathbf{G}) + P(\mathbf{F}) + R(\mathbf{G}) + R(\mathbf{F}) \\
&\quad = \sum_{i=1}^{m} \sum_{j=1}^{n} (\mathbf{E}_{ij}/\boldsymbol{\sigma}_{ij})^2 - \alpha \sum_{i=1}^{m} \sum_{h=1}^{p} \log \mathbf{G}_{ih} \\
&\qquad - \beta \sum_{h=1}^{p} \sum_{j=1}^{n} \log \mathbf{F}_{hj} + \gamma \sum_{i=1}^{m} \sum_{h=1}^{p} \mathbf{G}_{ih}^2 \\
&\qquad + \delta \sum_{h=1}^{p} \sum_{j=1}^{n} \mathbf{F}_{hj}^2. \qquad (6)
\end{aligned}$$

The coefficients α and β control the strength of two penalty terms which prevent the factors **G** and **F** from becoming negative. Similarly, γ and δ control two regularization terms which remove the rotational indeterminacy and control the scaling of the left and right factors. All the coefficients α, β, γ, δ are given smaller values during the iteration so that their final values are 'negligible but not zero'. For efficiency reasons, the logarithmic terms are approximated by suitable quadratic approximations which are updated before each step. — The penalty terms $P(\mathbf{G})$ and $P(\mathbf{F})$ are not needed in the unconstrained case but the regularization terms $R(\mathbf{G})$ and $R(\mathbf{F})$ are always necessary because they remove singularity (caused by rotations and scale changes) from the model.

2.8. What versus how in the algorithm of PMF

Before discussing the algorithm, it is essential to stress that the technical details of the algorithm should in fact not concern the user of PMF. In the case of PMF it is possible to separate the question 'What is computed? i.e. which mathematical problem is solved by the program?' from the question of how it is computed.

For the application the important aspects are in the first question: what are the connections between the physical model and the mathematical setup of the problem. The second aspect should only be important for the application if it is suspected that the program does not in fact do what it is supposed to do, that is in fact solving another mathematical problem.

This separation of what and how is not always possible. The DTD (direct trilinear decomposition, see Sanchez and Kowalski [7]) for computing the three-way PARAFAC factorization is an example: the mathematics is defined by defining an algorithm, and it is not at all obvious what the mathematical properties of the solution are.

In the present case the 'what' specification simply states that the program solves the problem of minimizing the (enhanced) object function Q as a function of the factor matrices **G** and **F**, obeying the non-negativity constraints. The links to the physical model are embedded in the non-negativity constraints, in the formulation of the main part of Q (especially in the values of the std-dev of **X**) and sometimes also in the regularization terms, especially if they are used in order to achieve a specific rotation.

2.9. The iterative PMF algorithm

Several aspects of the PMF model make it especially difficult to solve. The problem is non-linear because of the inequality constraints and also because the matrix of residuals depends on products of unknowns. The number of unknowns may be large, thousands or tens of thousands. This precludes the use of such algorithms where non-negativity is implemented by eliminating negative variables one variable at a time. And finally, the problem may be ill-posed: there are directions in the many-dimensional space of unknowns where the object function changes very slowly.

Because of these difficulties, the use of general purpose numerical packages to solve large PMF problems may not be practical and a specific algorithm 'PMF2' has been developed instead. The PMF2 algorithm may be considered as a generalization of the ALS algorithm.

When the algorithm is initially started from pseudorandom initial values the non-linearities are severe and simple means for computing the first steps are most cost-effective. At this stage the algorithm performs a few rounds of coordinate steps where one variable at a time is changed. These steps are simple to compute but they become quite inefficient when the iteration has proceeded to a more 'level' region of the many-dimensional space.

2.10. Computation of the main steps

During the main steps increments or ‘steps’ **g** and **f** are computed to the previous values **G** and **F** of the left and right factors. It is essential that both factors are changed together. The program minimizes the expression $Q(\mathbf{G}+\mathbf{g}, \mathbf{F}+\mathbf{f})$ with respect to **g** and **f**. It is possible to define the unknowns **g** and **f** as arbitrary matrices in the factor spaces **G** and **F**. Then there are $(m+n)p$ unknowns to be solved during each such ‘full’ step. This approach is used in the related program PMF3 for solving 3-way factor models. However, because of the very large numbers of unknowns in 2-way models solving the full model is not very efficient with respect to the computational workload. Thus, solving the full model is not used in the current version of PMF2 and instead the 2-way factor model is solved alternatingly in the following two restricted solution spaces:

First, the solution space is spanned by all components on the **G** side (mp unknowns), plus a single fixed direction on the **F** side. Altogether there are $mp+1$ unknowns to be determined. Second, the solution space is spanned by all components on the **F** side (np unknowns), plus a single fixed direction on the **G** side, altogether $np+1$ unknowns.

The extra terms (penalty, regularization) either are quadratic or else are approximated by quadratic expressions before each step is computed. Thus the main algorithm only deals with a quadratic object function. The extra terms cause a lot of extra bookkeeping in the program but they do not complicate the least squares fit part of the algorithm. In the following presentation only the main part of Q is shown, the extra terms are ignored.

The basic equations for the computation of a ‘full’ step are

$$(\mathbf{G}+\mathbf{g})(\mathbf{F}+\mathbf{f})=\mathbf{X}, \tag{7}$$

or

$$\mathbf{Gf}+\mathbf{gF}+\mathbf{gf}=\mathbf{R} \tag{8}$$

where **R** is the current residual, $\mathbf{R}=\mathbf{X}-\mathbf{GF}$. Eq. (8) may be simplified so that the second order term **gf** is omitted. This leads to Gauss–Newton steps, they are reliable but sometimes the convergence is not as fast as one could wish. Alternatively, the second order term **gf** may be kept in the computation, this leads to Newton–Raphson steps, faster convergence but sometimes a step fails with a non-positive-definite matrix. Both alternatives have been used on routine basis. We are currently experimenting with a compromise which promises the best of both but the results are not clear yet. The following presentation is based on the Gauss–Newton approach.

First consider the full step. Eq. (8) simplifies to the form

$$\mathbf{Gf}+\mathbf{gF}=\mathbf{R} \tag{9}$$

containing $(m+n)p$ unknowns $\mathbf{f}_{hj}$ and $\mathbf{g}_{ih}$. This matrix equation may be understood as a system of (mn) linear equations. The equations have the individual weights $(\mathbf{w}_{ij})$. In component form, one equation is

$$\sum_{h=1}^{p}\mathbf{G}_{ih}\mathbf{f}_{hj}+\sum_{h=1}^{p}\mathbf{g}_{ih}\mathbf{F}_{hj}=\mathbf{R}_{ij}\pm\boldsymbol{\sigma}_{ij}. \tag{10}$$

Eqs. (10) form an ordinary system of linear equations but they look strange because of the double indexing used for the unknowns. The equations could be assembled in the customary matrix form but the matrix is sparse, containing mostly zeroes, and the matrix presentation is clumsy in this case. The equations may be written in the general abstract form

$$\boldsymbol{\Gamma}\varphi=\mathrm{vect}(\mathbf{R}) \tag{11}$$

where the matrix $\boldsymbol{\Gamma}$ of dimensions $(mn\times(m+n)p)$ contains many zeroes and many repetitions of the values $\mathbf{G}_{ih}$ and $\mathbf{F}_{hj}$. The vector φ consists of all the $(m+n)p$ unknowns $\mathbf{f}_{hj}$ and $\mathbf{g}_{ih}$. The customary ‘normal equations’ (14) of the LS problem are formed in the usual way, according to the equations

$$\boldsymbol{\Psi}=\boldsymbol{\Gamma}^{T}\mathbf{W}\boldsymbol{\Gamma} \tag{12}$$

$$\gamma=\boldsymbol{\Gamma}^{T}\mathbf{W}\mathrm{vect}(\mathbf{R}) \tag{13}$$

$$\boldsymbol{\Psi}\varphi=\gamma, \tag{14}$$

where **W** is the diagonal matrix of the weights $\mathbf{w}_{ij}$. One should compute matrix $\boldsymbol{\psi}$ by using explicit formulae for its elements, not by straightforward matrix multiplication as suggested by the appearance of Eq. (12). Eq. (14) may be solved by Cholesky decomposition of the matrix $\boldsymbol{\psi}$. The inverse matrix $\boldsymbol{\psi}^{-1}$ is the covariance matrix for all the increments $\mathbf{f}_{hj}$ and $\mathbf{g}_{ih}$.

Now consider the computation of a ‘restricted’ step when in the F space the step is restricted to one direction (= matrix **f**) only. This matrix **f** is formed on the basis of preceding steps, based on the heuristic that the iteration generally continues in the same direction it has been proceeding during the previous steps. Eq. (9) gets the form

$$\alpha \mathbf{G}\mathbf{f} + \mathbf{g}\mathbf{F} = \mathbf{R} \tag{15}$$

where **f** is the fixed matrix (the step direction), α is a scalar F step length coefficient (to be determined), and **g** is the unknown matrix of increments to the matrix **G**. There are $mp+1$ unknowns. Eq. (10) gets the form

$$\alpha \mathbf{V}_{ij} + \sum_{h=1}^{p} \mathbf{g}_{ih}\mathbf{F}_{hj} = \mathbf{R}_{ij} \pm \boldsymbol{\sigma}_{ij} \tag{16}$$

where $\mathbf{V} = \mathbf{G}\mathbf{f}$ is a known matrix. The definitions of the symbols in Eqs. (11) to (14) change in a straightforward way from their original 'full step' definitions. The dimensions of the matrix $\boldsymbol{\Gamma}$ are now $(mn \times (mp+1))$. The matrix $\boldsymbol{\psi}$ is now essentially a block-diagonal matrix. It has only one full row in addition to the diagonal string of blocks. Computing the Cholesky decomposition of this sparse $\boldsymbol{\psi}$ is extremely fast, this is the main merit of using the restricted steps.

2.11. Computation of rotational substeps

Between the main Gauss–Newton or Newton–Raphson steps, rotational substeps are performed: the algorithm determines a rotation **T** and its inverse $\mathbf{T}^{-1}$ so that the new factor matrices **GT** and $\mathbf{T}^{-1}\mathbf{F}$ minimize the enhanced object function $\overline{Q}$. (Because the main part of $\overline{Q}$ does not change in rotations, this means that the rotation minimizes the sum of the expressions P and R in Eq. (6).) The algorithm PMF2 would also work without these rotational 'relaxation' steps but a speed gain of approximately a factor of 2 may be gained by also performing the rotational substeps.

3. Outliers and robust modelling

3.1. The nature of outliers

Three different possible reasons for outliers can be identified:

One observation does not belong to the set of the others. E.g. it may have some contamination. In precipitation studies, bird droppings or trapped insects are examples of this kind of outliers. Then a whole row of the matrix is (or may be) affected. If one is measuring uncontaminated terrain in order to determine the natural variability, and there is in fact one location with contamination, then one has this kind of outlier. In air pollution measurements a weak local source may be visible only occasionally, then it may be regarded as this kind of outlier. The famous July 4th observation, reported by Alpert and Hopke [8], also belongs to this class of outliers.

Secondly, one variable of one observation (= one element of the matrix **X**) may be in error or non-representative. This may happen because of a laboratory error (e.g. contamination or loss of analyte) or because of a typing/copying error. A contamination in the field may also produce this kind of outlier if the contaminant is a single compound. A peculiar local source could sometimes add a non-representative amount of some compound into an air-filter sample, it would not be a 'sampling error' but it would be an outlier.

It may happen that the values of some principal component coefficients ('scores') (more generally the values of any vector of the left or right factor matrices) obey a well-established distribution, e.g. normal, but one value (corresponding to one observation or to one variable) sticks out from this distribution. In other words, in one observation there is 'too little' or 'too much' of one factor component, but the composition of this component is not changed. In air pollution studies the so-called plumes (direct undiluted transport from source to receptor) may cause this kind of outlier. In this discussion we omit this third type of outliers because they do not cause a residual of the fit. It is in fact a question of definition if this type is considered an outlier or not.

3.2. Well-known examples of robust location estimators

The sample arithmetic mean is in fact a least squares estimator of population mean. It is well known that the mean is not robust, it is not safe to compute mean values of such data where outliers may occur. In contrast to this, the winsorized mean and the trimmed mean are robust estimators of the mean value (see Eadie et al. [9], pp. 184–187). Computation of these estimators is based on two cutting values l and u. The values of l and u are chosen so that a predetermined portion of the data values x_i are smaller than the lower cut value l and so that the same number of values are larger than u. The trimmed mean is simply the mean of only those values x_i which obey $l \leq x_i \leq u$. The outlying values are 'trimmed off'. The winsorized mean is computed from a modified set of values: each low-lying value $x_i \leq l$ is replaced by the low limit l, and similarly each high-lying value $x_i \geq u$ is replaced by the upper limit u. The arithmetic mean of this modified set is the winsorized mean of the original set. The trimmed mean and winsorized mean possess good statistical properties and in fact they should be used routinely in many kinds of environmental data analysis.

The median is a well known robust estimator. It may be visualized as the ultimate trimmed mean where all but one value have been trimmed off.

It is useful to see how these robust estimators may be computed by a weighted least squares fit.

3.3. The influence function of robust estimators

We rewrite Eq. (2) so that the dependence of the object function Q on individual data items is emphasized:

$$Q(\mathbf{E}) = \sum_{i=1}^{m} \sum_{j=1}^{n} Q_{ij}(\mathbf{E}_{ij}),$$

$$\text{where } Q_{ij}(\mathbf{E}_{ij}) = (\mathbf{E}_{ij}/\boldsymbol{\sigma}_{ij})^2 = \mathbf{w}_{ij}\mathbf{E}_{ij}^2. \qquad (17)$$

The $Q_{ij}(\ldots)$ should be understood as a functional that defines the dependence of Q on each residual $\mathbf{E}_{ij}$. Eq. (17) shows the formulation of $Q_{ij}(\ldots)$ for an ordinary weighted least squares fit. For unweighted least squares fit we have simply $Q_{ij}(\mathbf{E}_{ij}) = \mathbf{E}_{ij}^2$. Eq. (17) and other equations in this section may also be written with one summation (one subscript) for univariate data sets (e.g. computing a mean, X and E are vectors) or with triple summations (three subscripts) for 3-way factor analysis. Often the functional $Q_{ij}(\mathbf{E}_{ij})$ only depends on the value $\mathbf{E}_{ij}$ and not on the subscripts. Then we may simplify the notation and omit the subscripts from the Q.

It is simpler to work with the scaled residuals $\mathbf{R}_{ij} = \mathbf{E}_{ij}/\boldsymbol{\sigma}_{ij} = (\mathbf{X}_{ij} - \mathbf{Y}_{ij})/\boldsymbol{\sigma}_{ij}$ (the $\mathbf{Y}_{ij}$ are the fitted values, as defined in Eq. (1)). Thus Eq. (17) gets the form

$$Q(\mathbf{R}) = \sum_{i=1}^{m} \sum_{j=1}^{n} Q(\mathbf{R}_{ij}), \quad \text{where } Q(\mathbf{R}_{ij}) = \mathbf{R}_{ij}^2. \qquad (18)$$

It is customary to define the influence function $\boldsymbol{\Psi}(\ldots)$ as half of the derivative of the functional $Q(\ldots)$:

$$\boldsymbol{\Psi}(r) = 0.5\frac{\partial}{\partial r}Q(r), \qquad (19)$$

where r is a scaled residual. Combining Eqs. (18) and (19) gives the influence function for the usual least squares fit: $\boldsymbol{\Psi}(r) = r$.

The influence function may be visualized as the tension in a mechanical spring which connects the measured value and the mean value. In usual least squares the tension $\boldsymbol{\Psi}$ is proportional to the distance (residual) r: the farther away the data value is the stronger it pulls the fitted value (e.g. the mean) towards itself. This is non-robust behavior: a stray value may have an immense pull and thus it may ruin the correct model.

Various robust estimators may be constructed by specifying suitable functional forms for the influence function. Robustness is achieved by constructing 'soft' springs which limit the maximum tension to a reasonable value. If such a spring is stretched so that the tension is about to exceed this limit then the spring 'yields', i.e. the tension stays constant although the spring is stretched more and more.

It is easily seen that the following influence function defines the median:

$$\boldsymbol{\Psi}(r < 0) = -1, \qquad \boldsymbol{\Psi}(r > 0) = +1. \qquad (20)$$

This corresponds to the object function $Q(r) = 2|r|$.

The winsorized mean corresponds to the influence function

$$\boldsymbol{\Psi}(r < l) = l, \qquad \boldsymbol{\Psi}(l \leq r \leq u) = r,$$
$$\boldsymbol{\Psi}(r > u) = u \qquad (21)$$

The corresponding object function has a quadratic middle part between l and u with linear continuations towards plus and minus infinity.

3.4. Combining robustness and least squares

For computational reasons it is desirable to work with quadratic object functions when solving factor analytic problems. When an arbitrary object function Q is given, it may be possible to define an equivalent quadratic object function Q^{Q} so that these two functions are minimized at the same point. In this work we choose to work with the Huber influence function $\boldsymbol{\Psi}^{\mathrm{H}}$ defined by

$$\begin{aligned}\boldsymbol{\Psi}^{\mathrm{H}}(\mathbf{R}_{ij} < -\alpha) &= -\alpha,\\ \boldsymbol{\Psi}^{\mathrm{H}}(-\alpha \le R_{ij} \le \alpha) &= \mathbf{R}_{ij},\\ \boldsymbol{\Psi}^{\mathrm{H}}(\mathbf{R}_{ij} > \alpha) &= +\alpha,\end{aligned} \tag{22}$$

where $\mathbf{R}_{ij}$ is the scaled residual, see Huber [10,11]. The object function corresponding to $\boldsymbol{\Psi}^{\mathrm{H}}$ is denoted by Q^{H}. The user-selectable parameter α is analogous to the cutting limits of the winsorized mean, it determines the distance for classifying the observations as outlying: if the absolute value of the scaled residual $\mathbf{R}_{ij}$ is larger than α then the data point (ij) is an outlier. The outlying data are not rejected but they are handled as if they would be at the distance $\alpha\boldsymbol{\sigma}_{ij}$ from the fitted value.

A sufficient condition for the equivalence of the object functions Q^{Q} and Q^{H} is that the condition

$$\frac{\partial}{\partial \mathbf{R}_{ij}} Q^{\mathrm{Q}}_{ij}(\mathbf{R}_{ij}) = \frac{\partial}{\partial \mathbf{R}_{ij}} Q^{\mathrm{H}}_{ij}(\mathbf{R}_{ij}) \tag{23}$$

holds for all values (ij) when the derivatives are evaluated at the minimum of the selected Q^{H}. The definition of the influence function gives

$$2\boldsymbol{\Psi}^{\mathrm{H}}(\mathbf{R}_{ij}) = \frac{\partial}{\partial \mathbf{R}_{ij}} Q^{\mathrm{H}}_{ij}(\mathbf{R}_{ij}). \tag{24}$$

The desired quadratic object function is defined with the help of auxiliary coefficients $\mathbf{h}_{ij}$ (to be determined) as

$$\begin{aligned}Q^{\mathrm{Q}}(\mathbf{R}_{ij}) &= \sum_{i=1}^{m}\sum_{j=1}^{n}\left(\frac{\mathbf{R}_{ij}}{\mathbf{h}_{ij}}\right)^2\\ &= \sum_{i=1}^{m}\sum_{j=1}^{n}\left(\frac{\mathbf{X}_{ij}-\mathbf{Y}_{ij}}{\mathbf{h}_{ij}\boldsymbol{\sigma}_{ij}}\right)^2.\end{aligned} \tag{25}$$

Combining Eqs. (22), (23), (24), and (25) finally gives for each element (ij)

$$\frac{\mathbf{R}_{ij}}{\mathbf{h}^2_{ij}} = \boldsymbol{\Psi}^{\mathrm{H}}(\mathbf{R}_{ij}) = \begin{cases}-\alpha & \text{if} \quad \mathbf{R}_{ij} < -\alpha\\ \mathbf{R}_{ij} & \text{otherwise}\\ \alpha & \text{if} \quad \mathbf{R}_{ij} > \alpha\end{cases}. \tag{26}$$

The least squares approach to the robust factor analysis leads now to the equations

$$\begin{aligned}(\mathbf{G},\mathbf{F}) &= \underset{\mathbf{G},\mathbf{F}}{\arg\min} \sum_{i=1}^{m}\sum_{j=1}^{n} Q^{\mathrm{Q}}(\mathbf{R}_{ij})\\ &= \underset{\mathbf{G},\mathbf{F}}{\arg\min} \sum_{i=1}^{m}\sum_{j=1}^{n} \frac{(\mathbf{X}_{ij}-\mathbf{Y}_{ij})^2}{\mathbf{h}^2_{ij}\boldsymbol{\sigma}^2_{ij}},\end{aligned} \tag{27}$$

$$\mathbf{h}^2_{ij} = \begin{cases}1 & \text{if } |\mathbf{R}_{ij}| \le \alpha\\ |\mathbf{R}_{ij}|/\alpha & \text{otherwise}\end{cases} \tag{28}$$

where

$$\mathbf{Y}_{ij} = \sum_{k=1}^{p} \mathbf{G}_{ik}\mathbf{F}_{kj}.$$

The $\mathbf{h}_{ij}$ given by Eq. (28) depend on the fitted values $\mathbf{Y}_{ij}$. Thus Eq. (27) cannot be solved directly, without iteration. On the other hand, solving the model PMF is iterative in any case. Thus, it is natural to iterate so that whenever new values for $\mathbf{Y}_{ij}$ have been computed then new values for $\mathbf{h}_{ij}$ are computed from Eq. (28). These new $\mathbf{h}_{ij}$ are then used for computing the next set of values $\mathbf{G}$, $\mathbf{F}$, and $\mathbf{Y}$. There is no proof for the convergence of this iteration but in practice it has always converged without problems. However, the convergence of this modified iteration is often somewhat slower than the convergence of the basic PMF iteration.

It is seen that including the robust estimation principle into PMF is almost free. It amounts to computing increased effective std-dev values $\mathbf{h}_{ij}\boldsymbol{\sigma}_{ij}$ for all outlying data values after each iteration step. Otherwise the algorithm is not changed, it is still a least squares computation. A typical choice for the cutoff parameter would be $\alpha = 4$.

3.5. Scaling the rows versus reweighting the individual observations

For a review of earlier robust factor analytic techniques see Singh [12]. The least squares approach has not been used in earlier work, thus it has not been possible to treat individual data values as outliers in the way now described for PMF. Instead, the handling has been based on individual observations, i.e. individual matrix rows. If the Mahalanobis distance for a row exceeds a threshold value α then the whole row is weighted down. This is possible by left-multiplying the matrix **X** with a diagonal matrix whose entries are the desired weight factors. Such a scaling transform is compatible with the Eigenanalysis, see Paatero and Tapper [1].

No comparison has yet been made between the two approaches. The following is presented as a plausible conjecture without proof:

''For the outliers of the first kind (the whole observation deviates from the rest) the traditional technique could well be the best. The indication for a row being outside is best obtained by considering the whole rows, and all the values should be punished collectively on such a row.''

''For the outliers of the second kind (individual data values in error) the traditional approach would undoubtedly detect the situation if the error is large enough. However, smaller errors could go unnoticed because they are masked by the fluctuations of the other values on the row. Also, the traditional approach would cause more loss of information than necessary because the whole row would need to be rejected if there is one bad value.''

An extreme case would be a large matrix where on each row one or more individual values are in error. The traditional approach would need to reject all rows! The example shown below demonstrates that the PMF approach is able to solve such a case without problems.

4. Producing desired rotations

There are several possibilities for producing a solution with desired rotational properties. It is possible to rotate the result computed by PMF either by hand or programmatically. The drawbacks of this approach have been discussed by Paatero and Tapper [2].

The user may fix selected elements of the factor matrices **G** and/or **F** to zero before starting PMF. In this way the solution is forced to rotate to such a position where the selected elements are zero — this position may or may not be unique. However, this technique is not fully objective. Another drawback is that fixing some elements to zero seems to increase the likelihood of occurrence of local minima of the enhanced function Q, leading to multiple solutions of the problem.

It is also possible to add extra terms in the object function which is to be minimized by PMF. These terms represent 'target shapes' which may be formulated so that they cause the program to favor such rotations which add together factors on the F side and at the same time subtract factors on the G side. The opposite is also possible, favoring additions on G side and subtractions on F side. This technique is used in the current program PMF2, it leads to a complicated mathematical analysis and is not described now. The example presented in the following section was calculated by this technique.

Finally it is possible to modify the object function so that different mathematical expressions are used for the regularization terms on F and G sides. The following approach appears promising although we cannot yet quote specific results. Define the regularization terms $R(\mathbf{G})$ and $R(\mathbf{F})$ as

$$R(\mathbf{G}) = 2\delta \sum_{i=1}^{m} \sum_{h=1}^{p} |\mathbf{G}_{ih}| / \gamma_h,$$

$$R(\mathbf{F}) = \delta \sum_{h=1}^{p} \sum_{j=1}^{n} \mathbf{F}_{hj}^2 \tag{29}$$

where the scaling constants γ_h are computed before each iteration step as $\gamma_h = \sum_{i=1}^{m} |\mathbf{G}_{ih}|$ (their role is to impose unit Euclidean length to the F factors). The value of δ controls the strength of the regularization. Straightforward analysis shows that this non-symmetric form of regularization favors subtractions of factors on the G side and additions on the F side. Thus this technique should reproduce the results shown in the example. Of course, the regularization may also be defined in the opposite way, favoring additions on the G side.

5. The analysis of a synthetic test case with and without outliers

There are conflicting aspects in selecting test/demonstration cases in factor analysis. The ultimate test of factor analytic software is analyzing real data. Especially for robust techniques, environmental data might offer good test cases. However, real data contain surprises and one does not always know what is the 'correct result'. Also, real data from one field is usually not instructive for researchers in other fields. For these reasons it is advisable to test factor analytic techniques with well-documented simulated examples, such as the 'Quail Roost II' test data for 2-way factorization and mass balance calculations, see Currie et al. [13].

Environmental data, even if simulated, makes difficult demonstration cases because there are no clean visual patterns in the data. The values appear random to the eye and when comparing different analyses of the same data it is not easy to see where a change has been made. The experience in developing factor analytic algorithms has shown that test matrices should look nice when plotted, so that the human eye may spot changes in the factors easily. There should be both large and small values in the factors. For these reasons a simulated spectroscopic-like example was chosen for demonstrating the technique PMF in this paper. Outliers are also demonstrated with this example although they are probably not very common in real spectroscopic studies. The example is based on analytic shapes (Gaussian, exponential) but these analytic properties are not utilized by the programs. This example does not utilize the ability of the program to handle individual std-dev for individual data values.

5.1. The Gauss-exponential test matrix

The matrix of size 40 × 20 contains four factors but it has been constructed so that it is not far from being singular. The shapes of factors are analytic (G is Gaussian, F is exponential). The shapes of correct factors are such that there is rotational freedom in the solutions. However, the desired 'pure-peak' solution is at the extreme edge of the rotational domain, thus by using the correct rotational option one may in fact obtain the desired solution automatically, as seen in the results. The elements of matrix **X** are defined by

$$\mathbf{X}_{ij} = \sum_{h=1}^{4} a_h e^{-0.5(i-u_h)^2/w_h^2} e^{-jt_h} + r_{ij}, \tag{30}$$

the numerical values of the parameters are given in Table 1.

Table 1
The parameters of the four Gaussian/exponential factor components in the simulated example

Parameter	Number of component			
	1	2	3	4
a	100	50	150	50
u	10	20	27.7	32.5
w	2.5	2.5	2.5	2.5
t	0.2	0.35	0.6	0.07

Version 1. No outliers, $\mathbf{X}_{ij}$ contains normally distributed pseudorandom error r_{ij} with std-dev = 0.01. The first five singular values of **X** (vers. 1) are 76, 18, 7.2, 0.5, 0.09.

Version 2. With outliers: **X** contain white noise r_{ij} as above, but in addition a randomly selected 10% of $\mathbf{X}_{ij}$ contain normally distributed pseudorandom error with std-dev = 0.2. The first singular values of **X** (vers. 2) are 76, 18, 7.3, 0.86, 0.69. It is seen that the outlier noise dominates over the original fourth sin-

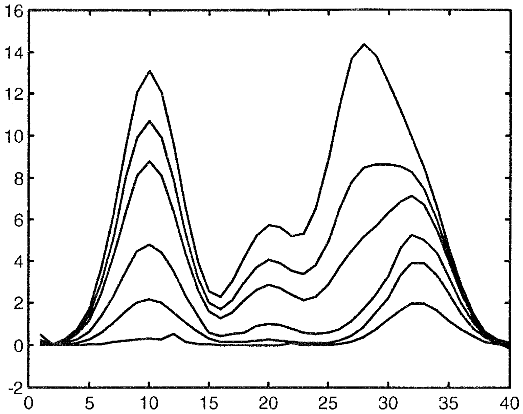

Fig. 1. The outlier-containing synthetic matrix **X** (version 2) of dimensions 40 × 20, used for example factorizations. The columns of the matrix contain 'spectra' consisting of Gaussian peaks. Only the columns 1, 2, 3, 6, 10, and 20 are shown in the diagram.

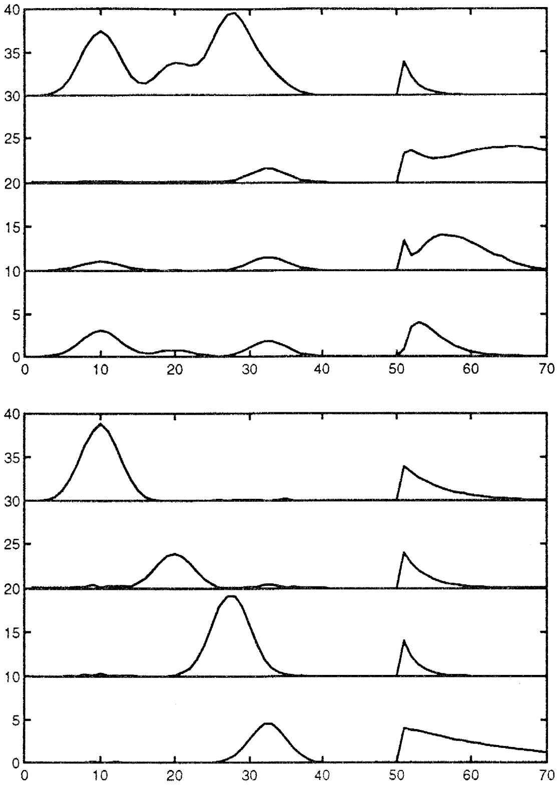

Fig. 2. Factorization results for the matrix **X** (version 1, no outliers). Each of the four levels of the figure shows one computed factor. On the left, in positions 1 to 40, the G factor ('spectrum') is shown. In positions 51 to 70 is the corresponding F factor, the 'decay curve'. (a, top) No specific rotation was attempted. (b, bottom) The factorization was performed so that subtractions were favored on the G side and additions on the F side.

gular component, thus it is expected that non-robust techniques will not resolve all four components.

In Fig. 1 a few of the 'spectra' = columns (1, 2, 3, 6, 10, 20) of the matrix **X** (version 2) are shown. The outliers can be seen in a few places. The four 'peaks' decay exponentially with individual half-lives.

5.2. Results: no outliers in the matrix

Fig. 2a shows the raw results of PMF2. On each of the four levels of the diagram, one factor is represented. On the left, in positions numbered 1 to 40, the column vectors of the left factor matrix **G** are shown. Ideally these curves should show single Gaussians. On the right, in positions numbered 51 to 70 the corresponding row vectors of the **F** matrix are shown, ideally they should be exponential decay curves. It is seen that the original correct factors are not recovered, the G factors are not single Gaussian peaks and the F factors are not exponential curves. Instead, the results are a rotation of the desired 'true solution'. On the G side we may discern that the true single-peak factors have been added to each other. Similarly, on the F side the exponential curves have been subtracted from each other. This suggests that the opposite transforms (subtractions for G, additions for F) might be able to produce 'better' solutions. The program was rerun using automatically generated target functions for causing subtractions for G, additions for F. Indeed the results were now almost pure Gaussian peaks on the G side and almost pure exponentials on the F side, as shown in Fig. 2b.

There is surprisingly little noise in these results, considering that the signal-to-noise distance between the smallest 'true' singular value and the largest noise singular value is only the ratio of 0.5 to 0.09, a factor of 5.5. There is no explanation for this at the moment.

The ability to recover the true solution by this automatic rotation technique depends critically on the fact that each true G factor contains many values which are close to zero or exactly zero. In other words, the desired true solution is situated at the extreme end of the domain which is accessible by rotations. If the true shapes would be exponentials on both sides then it would not be possible to recover them by automatic rotational techniques. The question of recovering correctly rotated factor shapes automatically with a program has also been studied by Henry [14,15]. His approach 'SAFER' (source apportionment by factors with explicit restrictions) also depends on the presence of zeros in the correct factor matrix.

5.3. Results: outliers in the matrix, non-robust analysis

Fig. 3a and b show similar results as the preceding figures, but now computed using a non-robust algorithm on the outlier-containing version 2 of the test matrix. As expected, the fourth factor is not seen. In Fig. 3a there appears to be less noise because all four factor diagrams are mostly based on the three well-resolved singular components. When the rotations attempt to concentrate each true factor to one computed curve the lack of information becomes apparent in Fig. 3b.

5.4. Results: outliers in the matrix, robust analysis

Fig. 4a and b show similar results as the preceding figures, but now computed using a *robust algorithm* (PMF2 with Huber influence function, $\alpha = 2$) on the outlier-containing version 2 of the test matrix. It is seen that the noise is well suppressed, four factors are well recovered. It would be interesting to see how customary robust techniques would handle this case. On the average there are two outlying values on each row of the matrix. Thus almost all rows would need to be weighted down by the customary row-weighting technique and it seems unlikely that row-weighting techniques could succeed with this matrix. However, it might be possible to succeed with such techniques which iteratively replace outlying observed values $\mathbf{X}_{ij}$ with the corresponding fitted values $\mathbf{Y}_{ij}$.

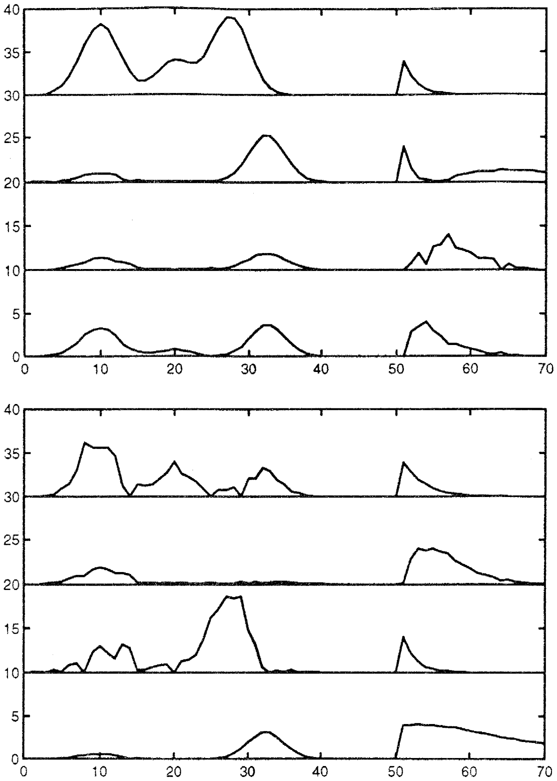

Fig. 3. Factorization results for the matrix **X** (version 2, outliers in 10% of the matrix elements). The program PMF2 was run in *non-robust* mode, without iterative reweighting. (a, top) No specific rotation was attempted. (b, bottom) The factorization was performed so that subtractions were favored on the G side and additions on the F side.

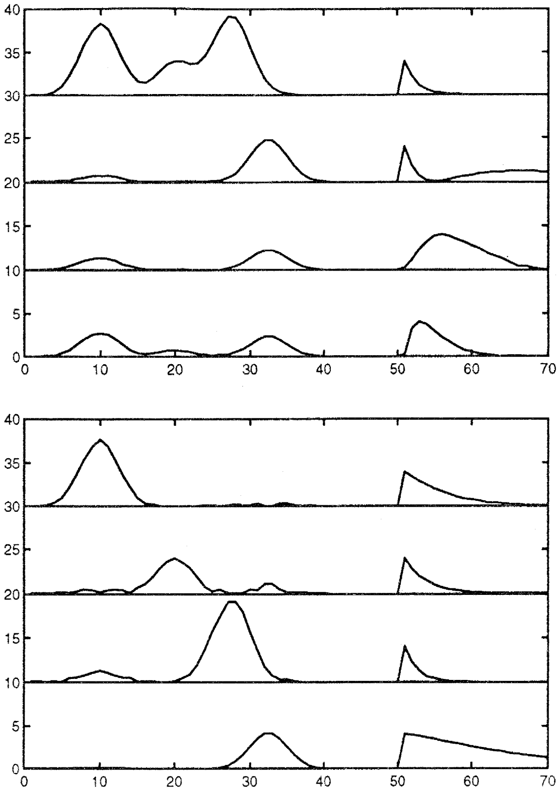

Fig. 4. Factorization results for the matrix **X** (version 2, outliers in 10% of the matrix elements). The program PMF2 was run in *robust* mode, with iterative reweighting of outlying values ($\alpha = 2$). (a, top) No specific rotation was attempted. (b, bottom) The factorization was performed so that subtractions were favored on the G side and additions on the F side.

6. Availability of the programs

The programs PMF2 and PMF3 have been written in the language Fortran90. At present they have been compiled for 386–486-Pentium PC computers and for the DEC Alpha processors. The .exe files are available from the author for a free trial period of six months.

References

[1] P. Paatero and U. Tapper, Chemom. Intell. Lab. Syst. 18 (1993) 183–194.

[2] P. Paatero and U. Tapper, Environmetrics 5 (1994) 111–126.

[3] S. Juntto and P. Paatero, Environmetrics 5 (1994) 127–144.

[4] P. Anttila, P. Paatero, U. Tapper and O. Järvinen, Atmos. Environ. 29 (1995) 1705–1718.

[5] R.A. Harshman and M.E. Lundy, in: Research Methods for Multimode Data Analysis, Law et al. (Eds.) (Praeger, New York, 1984) pp. 122–215.

[6] R.T. Ross and S. Leurgans, Methods Enzymol. 246 (1995) 679–700.

[7] E. Sanchez and B.R. Kowalski, J. Chemom. 4 (1990) 29–45.

[8] D.J. Alpert and P.K. Hopke, Atmos. Environ. 15 (1981) 675–687.

[9] W.T. Eadie, D. Drijard, F.E. James, M. Roos and B. Sadoulet, Statistical Methods in Experimental Physics (North-Holland, Amsterdam, 1971).

[10] P.J. Huber, Ann. Math. Stat. 35 (1964) 73–101.

[11] P.J. Huber, Robust Statistics (John Wiley, New York, 1981).

[12] A. Singh, in: Multivariate Environmental Statistics, G.P. Patil and C.R. Rao (Eds.) (Elsevier Science Publishers, Amsterdam, 1993) pp. 445–488.

[13] L.A. Currie, R.W. Gerlach, C.W. Lewis, W.D. Balfour, J.A. Ooper, S.L. Dattner, R.T. De Cesar, G.E. Gordon, S.L. Heisler, P.K. Hopke, J.J. Shah, G.D. Thurston and H.J. Williamson, Atmos. Environ. 18 (1984) 1517–1537.

[14] R.C. Henry, in: Receptor Modeling for Air Quality Management, P.K. Hopke (Ed.) (Elsevier, New York, 1991) pp. 117–148.

[15] R.C. Henry and B.M. Kim, Chemom. Intell. Lab. Syst. 8 (1990) 205–216.