JOURNAL OF CHEMOMETRICS
*J. Chemometrics* **12**, 223–247 (1998)

# LEAST SQUARES ALGORITHMS UNDER UNIMODALITY AND NON-NEGATIVITY CONSTRAINTS

RASMUS BRO[1]* AND NICHOLAOS D. SIDIROPOULOS[2]
[1]*Chemometrics Group, Food Technology, Department of Dairy and Food Science, Royal Veterinary and Agricultural University, DK-1958 Frederiksberg C, Denmark*
[2]*Department of Electrical Engineering, University of Virginia, Charlottesville, VA 22903, U.S.A.*

SUMMARY

In this paper a least squares method is developed for minimizing $||\mathbf{Y} - \mathbf{X}\mathbf{B}^T||_F^2$ over the matrix $\mathbf{B}$ subject to the constraint that the columns of $\mathbf{B}$ are unimodal, i.e. each has only one peak, and $||\mathbf{M}||_F^2$ being the sum of squares of all elements of $\mathbf{M}$. This method is directly applicable in many curve resolution problems, but also for stabilizing other problems where unimodality is known to be a valid assumption. Typical problems arise in certain types of time series analysis such as chromatography or flow injection analysis. A fundamental and surprising result of this work is that unimodal least squares regression (including optimization of mode location) is not any more difficult than two simple Kruskal monotone regressions. This had not been realized earlier, leading to the use of either undesirable *ad hoc* methods or very time-consuming exhaustive search algorithms. The new method is useful in and exemplified with two- and multi-way methods based on alternating least squares regression solving problems from fluorescence spectroscopy and flow injection analysis. © 1998 John Wiley & Sons, Ltd.

KEY WORDS: unimodal least squares regression; alternating least squares regression; PARAFAC; PARATUCK2; curve resolution; chromatography; flow injection analysis; fluorescence spectroscopy; non-negativity; oligomodal

## 1. INTRODUCTION

In curve resolution it is quite common to work with data types where the underlying phenomena generating the data can be assumed to be unimodal. As an example consider a matrix $\mathbf{X}$ containing in its rows UV-vis spectra of a sample measured at different times after injection in a chromatographic column, each row representing the spectrum at a certain time. If the sample contains $F$ different spectrally active analytes and no baseline is present, it is theoretically valid to describe the $I \times J$ matrix $\mathbf{X}$ by a bilinear model

$$\mathbf{X} = \mathbf{A}\mathbf{D}^T + \mathbf{E} \tag{1}$$

where $\mathbf{A}$ is an $I \times F$ matrix, $\mathbf{D}$ is a $J \times F$ matrix and $\mathbf{E}$ is the unmodeled residual part of $\mathbf{X}$. In

* Correspondence to: R. Bro, Chemometrics Group, Food Technology, Department of Dairy and Food Science, Royal Veterinary and Agricultural University, DK-1958 Frederiksberg C, Denmark. E-mail: rb@kvl.dk
Contract/grant sponsor: Nordic Industry Foundation; Contract/grant number: P93149.
Contract/grant sponsor: FØTEK.
Contract/grant sponsor: National Science Foundation; Contract/grant number: NSF EEC 9402384.
Contract/grant sponsor: Lockheed-Martin Chair in Systems Engineering.

CCC 0886–9383/98/040223–25 $17.50

*Received 12 June 1997*
*Accepted 6 February 1998*

expanded form the model is

$$x_{ij} = \sum_{f=1}^{F} a_{if} d_{jf} + e_{ij}. \tag{2}$$

If **D** contains in its *f*th column the spectrum of the *f*th analyte, then the *f*th column of **A** will be the corresponding chromatogram of that analyte. If the chromatographic analysis is working well, the chromatographic profiles can be assumed to be unimodal.

In multi-way analysis, similar problems to the above-mentioned chromatographic problem can easily be envisioned by e.g. measuring several different samples in the same fashion as above. The data will be a three-way array of size $I \times J \times K$, i.e. $I$ samples each measured spectrophotometrically $K$ times at $J$ wavelengths. If Beer's law is assumed to hold and every analyte has the same chromatographic profile in every run, then the data can be approximated by a trilinear model

$$x_{ijk} = \sum_{f=1}^{F} a_{if} d_{jf} c_{kf} + e_{ijk} \tag{3}$$

stating that the absorbance $x_{ijk}$ of the $i$th sample at the $j$th wavelength at time $k$ will be the sum of contributions from each of the $F$ analytes present in the samples. For each analyte $f$ in a sample $i$ the contribution is modeled as the initial concentration of analyte, $a_{if}$, times the absorptivity of that analyte at wavelength $j$, $d_{jf}$, times the fraction of the analyte present at the detector at time $k$, $c_{kf}$. The noise part of $x_{ijk}$ is called $e_{ijk}$. This model is one possible extension of the PCA (principal component analysis) model to higher-order data and is called the PARAFAC (parallel factor analysis) model.[1]

In bilinear modeling there is a well-known problem of rotational freedom, thus necessitating further constraints to be imposed to identify the model. In PCA one imposes the constraints that loading and score vectors, i.e. the components, are orthogonal. In curve resolution these kinds of constraints are useless, as the underlying phenomena sought (spectra, profiles, etc.) are by no means orthogonal in general. To obtain unique models that reflect the pure underlying spectra, the key is to find selective or partly selective channels where some analytes are not present. This knowledge can be used to obtain unique models. Other typical constraints or restrictions used are non-negativity and requiring unimodality of chromatographic profiles. In three-way modeling the problem of identification vanishes if the data are trilinear. Apart from scaling and permutations of components, the trilinear model is unique under mild conditions.[2–5] Hence the pure spectra, pure chromatograms and pure concentrations will be found up to a scaling.

Some three-way data sets are, however, still difficult to estimate even though the trilinear model is theoretically an appropriate model. Sampling variation, noise and very similar profiles can cause the model to be impossible or difficult to estimate reliably. The three-way structure of the model in these cases is not sufficient information in itself to ensure meaningful estimates. Incorporating sensible constraints will help in getting the valid information from the data, as shown in a similar context in Reference 6.

Least squares unimodal regression is important, as currently either *ad hoc* or very restrictive methods are used in chemometrics for enforcing unimodality.[7–10] One approach often used in iterative algorithms is to simply change elements corresponding to local maxima on an estimated curve so that the local maxima disappear. Clearly such a method does not have any least squares or other well-defined properties. The use of such methods, especially as a substep in a larger optimization algorithm, can be problematic, as it might cause the overall algorithm to diverge instead of converge to a solution. The restrictive methods typically enforce the profiles to be Gaussians, but there is seldom provision for assuming that e.g. chromatographic peaks are even approximately

 

Gaussian. Least squares estimation under unimodality constraints seems to be more appropriate than the overly restricted Gaussian approach and more well-defined and well-behaved than simply changing parameters without considering the accompanying changes in the loss function. In other words, enforcing unimodality is expected to be a strong enough restriction for unscrambling difficult data, yet flexible enough to avoid over-restricting the model, as unimodality is often closer to what is presumed to be appropriate than requiring Gaussian profiles.

It is our aim in a broader context to develop sufficient theory to formulate a curve resolution problem as *one* global problem stated as a structural model with constraints. Most curve resolution methods do not attempt to solve one global optimization problem.[11,12] A typical set-up is to estimate the profile and spectrum of the analyte having the most selective window, e.g. a subset of variables where only the current analyte seems to be present. The estimated spectrum and profile are then subtracted from the data and the next analyte is estimated from this new data set. Estimating a model with one well-defined optimization criterion is believed to be stabilizing in situations where traditional algorithms fail to give meaningful results. For this purpose it is important to be able to state technological and chemical *a priori* knowledge in a concise mathematical language that enables rigorous incorporation of such knowledge in the specific model to be estimated. Apart from unimodality, non-negativity of parameters,[13,14] equality of parameters, smoothness of e.g. spectral estimates, allowing for closure, selectivity[11,12] and fixing parameters are important general constraints that one should be able to incorporate specifically into a model. Most of these constraints can be incorporated into least squares algorithms using results from numerical analysis and other mathematical sciences.

### 1.1. Organization

The rest of this paper is organized as follows. In the next subsection we present some key background material. In Section 2 we define the problem. In Section 3 we develop and prove the correctness of an algorithm for unimodal least squares regression. We also discuss several possible modifications of the algorithm. In Section 4 we give experimental proof of the usefulness of the algorithm by showing one simulated and two real examples of its use. The Appendix contains the collected proofs of several lemmas and theorems.

Scalars, including elements of vectors and matrices, are indicated by lower-case italics and vectors by bold lower-case characters; bold capitals are used for two-way matrices and underlined bold capitals for three-way arrays. The letters $I$, $J$ and $K$ are reserved for indicating the dimensions of the first, second and third modes of a three-way array respectively and $i$, $j$ and $k$ are used as indices for each of these modes. An $I \times J \times K$ array $\underline{\mathbf{X}}$ is also equivalently named $x_{ijk}$, implicitly assuming that the indices run from one to the respective dimensionalities. A subarray of a three-way array $\underline{\mathbf{X}}$ is called $\mathbf{X}_k$ if it is the $k$th $I \times J$ layer in the third mode. An $I \times J$ matrix $\mathbf{X}$ is sometimes referred to by its column vectors as $[\mathbf{x}_1\ \mathbf{x}_2 \ldots \mathbf{x}_J]$.

### 1.2. Background

Most methods used for estimating the PARAFAC model are based on alternating least squares (ALS). The principle behind ALS is to divide the parameters into several sets. Each set of parameters is estimated in a least squares sense conditionally on the other parameters. The estimation of parameters is repeated iteratively until no significant change is observed in the parameter values or in the fit of the model to the data. It is easy to see that such an algorithm always converges in a feasible set. As all estimations of parameters are least squares estimations, such an algorithm may only improve the fit or keep it the same. Since the problem is a bounded cost problem, convergence follows. In many cases the overall problem may have several local minima, which means that convergence to the global

 

optimum can seldom be guaranteed but will be dependent on data, model and algorithm. While some ALS algorithms, e.g. NIPALS[15] for estimating a principal component analysis model or most algorithms for estimating the Tucker3 $N$-mode principal component analysis model,[16] are very fast and stable, most algorithms for estimating e.g. the PARAFAC model *can* occasionally be problematic for certain types of data.[17] It is, however, frequently observed that as long as the model is well suited for the data at hand, convergence to a global minimum is usually achieved. Simple repetitions of the analysis can reveal if global convergence has not been achieved, as convergence to the same *local* optimum several consecutive times is unlikely if the analysis is started from different initial parameter sets.

For the two-way problem in equation (1) a very simple ALS algorithm could be the following.

0. Initialize $\mathbf{A}$.
1. $\mathbf{D} = \mathbf{X}^T\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}$.
2. $\mathbf{A} = \mathbf{X}\mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1}$.
3. Go to step 1 until convergence.

In steps 1 and 2 one may use pseudo-inverses. In step 1 of the above algorithm $\mathbf{D}$ is the solution of minimizing

$$\min_{\mathbf{D}} ||\mathbf{X} - \mathbf{A}\mathbf{D}^T||_F^2 \tag{4}$$

and in step two $\mathbf{A}$ minimizes a similar objective function.

For PARAFAC a similar tentative algorithm can be given. Let $\mathbf{A}$ be an $I \times F$ matrix holding the parameters $a_{if}$ and let $\mathbf{D}$ ($J \times F$) and $\mathbf{C}$ ($K \times F$) be defined likewise. Then the PARAFAC algorithm is as follows.

0. Initialize $\mathbf{D}$ and $\mathbf{C}$.
1. $\mathbf{A} = \mathbf{X}^T\mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}$.
2. $\mathbf{D} = \mathbf{X}^T\mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}$.
3. $\mathbf{C} = \mathbf{X}^T\mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}$.
4. Go to step 1 until convergence.

The matrices $\mathbf{X}$ and $\mathbf{Z}$ are working matrices that are redefined in every step. The matrix $\mathbf{X}$ contains the three-way array $\underline{\mathbf{X}}$ rearranged in matrix form, while the matrix $\mathbf{Z}$ is defined through the parameters not being estimated. For estimating $\mathbf{C}$, the matrix $\mathbf{Z}^T\mathbf{Z}$ ($F \times F$) can be calculated as

$$\mathbf{Z}^T\mathbf{Z} = (\mathbf{A}^T\mathbf{A}) \circ (\mathbf{D}^T\mathbf{D})$$

the operator $\circ$ being the Hadamard (element-wise) product.[18] The matrix $\mathbf{X}^T\mathbf{Z}$ can be calculated as

$$\mathbf{X}^T\mathbf{Z} = \begin{bmatrix} \text{diag}(\mathbf{A}^T\mathbf{X}_1\mathbf{D})^T \\ \text{diag}(\mathbf{A}^T\mathbf{X}_2\mathbf{D})^T \\ \vdots \\ \text{diag}(\mathbf{A}^T\mathbf{X}_K\mathbf{D})^T \end{bmatrix}$$

where diag($\mathbf{G}$) is a column vector containing the diagonal elements of $\mathbf{G}$. For further details on the PARAFAC model and algorithm see References 4, 5 and 19. The important aspect here is that the objective in each step, e.g. step 3, is of the form

$$\min_{\mathbf{C}} ||\mathbf{X} - \mathbf{Z}\mathbf{C}^T||_F^2 \tag{5}$$

It is easily seen that estimating one set of parameters in either the two-way, three-way or $N$-way

 

problems in general amounts to the same thing. If this estimate is sought under the constraint that the columns of **C** are unimodal, we will call it problem **UNIMODAL**.

In general the optimization problem is as follows. Given **Y** ($I \times J$) and **X** ($I \times F$),

$$\text{minimize } ||\mathbf{Y} - \mathbf{X}\mathbf{B}^{\mathrm{T}}||_{\mathrm{F}}^2$$
$$\text{subject to } \textit{columns of } \mathbf{B} \textit{ are unimodal}$$

and optionally the elements of **B** are non-negative. To facilitate further discussion, we will initially show that problem **UNIMODAL** can be partitioned into a set of simpler problems by means of an ALS approach, where each column of **B** is estimated given the remaining columns. Let $\mathbf{B}^{(-f)}$ be the $J \times (F-1)$ matrix consisting of all but the $f$th column of **B**. Let $\mathbf{X}^{(-f)}$ be the $I \times (F-1)$ matrix consisting of all but the $f$th column of **X**, and $\mathbf{x}_f$ the $f$th column of **X**. Let $\mathbf{Y}^{(-f)}$ equal $\mathbf{Y} - \mathbf{X}^{(-f)}\mathbf{B}^{(-f)}$. For the $f$th column of **B**, called $\mathbf{b}_f$, the problem to be solved is then

$$\text{minimize}||\mathbf{Y}^{(-f)} - \mathbf{x}_f\mathbf{b}_f)^{\mathrm{T}}||_{\mathrm{F}}^2$$
$$\text{subject to } \mathbf{b}_f \textit{ is unimodal}$$

An ALS algorithm for solving problem **UNIMODAL** can then be written.

0. Initialize **B**.
1. For every $f$ (1 to $F$) estimate the $f$th column of **B** as the solution to the minimization problem above.
2. Update **B** and go to step 1 until convergence.

The convergence of this algorithm follows from the convergence of ALS algorithms. From this it follows that it is sufficient to obtain a solution to the simpler problem of estimating one column of **B** at a time. The solution to this problem will be described in the next section.

## 2. DEFINING THE PROBLEM

Our unimodal least squares regression (**ULSR**) problem can now be stated as follows.

**Problem 1**

(**ULSR**) Given an $I \times J$ matrix **Y** and an $I \times 1$ vector **x**,

$$\text{minimize}||\mathbf{Y} - \mathbf{x}\mathbf{b}^{\mathrm{T}}||_{\mathrm{F}}^2$$
$$\text{subject to } \mathbf{b} \textit{ is unimodal}$$

We are particularly interested in non-negative **ULSR** (note that a bounded problem can always be transformed to a non-negative problem), in which case the unimodality constraint can be expressed as

$$b_1 \geq 0$$
$$b_j \geq b_{j-1},\ j = 2, \ldots, n$$
$$b_J \geq 0$$
$$b_j \leq b_{j-1},\ j = n+1, \ldots, J = \text{ size}(\mathbf{b})$$

for some mode location $n$, $1 \leq n \leq J$, which *is itself subject to optimization*. In the sequel, whenever we say **ULSR**, we mean non-negative **ULSR**. We have the following important lemma.

 

**Lemma 1**

Let $\boldsymbol{\beta}$ be the *unconstrained* LS solution to the problem of minimizing $||\mathbf{Y} - \mathbf{x}\mathbf{b}^T||_2^2$. Then the **ULSR** Problem 1 above is equivalent to

$$\text{minimize} ||\boldsymbol{\beta} - \mathbf{b}||_F^2$$
$$\text{subject to } \mathbf{b}\text{: } \textit{unimodal}$$

This result is not dependent on the type of constraint but pertains to all constrained regression problems where the restrictions of the parameters are independent of remaining parameters. This has been shown and utilized in several similar settings, e.g. Reference 20. A brief proof of Lemma 1 can be found in the Appendix.

## 3. UNIMODAL LEAST SQUARES REGRESSION

It is not difficult to envision an algorithm for solving problem **ULSR**. This can for example be done using a quadratic programming as will be shown later. However, when used in iterative algorithms and in ALS algorithms in particular, speed is of utmost importance. In fact, several authors have proposed algorithms for solving problem **ULSR**, though not in the context of problem **UNIMODAL**. We will describe the relation between these algorithms and ours after having described the algorithm.

As expected, unimodal regression is related to monotone regression. The basic principle underlying monotone regression will be outlined and an algorithm for unimodal regression will be developed. We will do this by first describing an algorithm for solving the problem for fixed mode location and then show how this algorithm can be modified for solving the general problem. Detailed information on monotone regression is given in References 21–23.

Consider a $J \times 1$ vector $\boldsymbol{\beta}$ with typical element $\beta_j$. A vector $\mathbf{b}$ is sought that minimizes the sum-squared difference between $\boldsymbol{\beta}$ and $\mathbf{b}$ subject to the requirement that $\mathbf{b}$ is monotone increasing, i.e. $b_j \leq b_{j+1}$. We will only consider the situation where all elements of $\mathbf{b}$ are free to attain any value whatsoever. This is a situation with no ties according to Kruskal.[22] Consider two consecutive elements $\beta_j$ and $\beta_{j+1}$. Suppose $\beta_{j+1} < \beta_j$, then what should the values of $\mathbf{b}$ be to give the best monotone estimate of $\boldsymbol{\beta}$? If no other elements are violating the constraints implied by the monotonicity, then all elements except the $j$th and the $(j+1)$th should equal the corresponding elements of $\boldsymbol{\beta}$, as this will naturally lead to a zero contribution to the sum-squared error. It further holds that the elements $b_j$ and $b_{j+1}$ should be set to the mean of $\beta_j$ and $\beta_{j+1}$ (for simplicity assuming that the mean is higher than $b_{j-1}$ and lower than $b_{j+2}$). From the geometry of the problem it follows that any other set of values will produce a higher sum-squared error. This observation is the cornerstone of monotone regression.

Define a *block* as a set of consecutive elements of $\mathbf{b}$ all having been assigned the same value. Initially let $\mathbf{b}$ equal $\boldsymbol{\beta}$ and let every element of $\mathbf{b}$ be a block. Let the first leftmost block be the *active* block. If the common value of elements of the active block is higher than or equal to the common value of the block to the left, the block is *down-satisfied;* otherwise concatenate the two blocks into one block whose elements have a common value equal to the mean of the elements of the two blocks. If the new common value of the block is lower than or equal to the value of the block to the right, the block is *up-satisfied;* otherwise the two blocks are averaged. Continue checking up- and downwards until the resulting block is both up- and down-satisfied, then continue to the next block, i.e. the next block becomes active. When the last block is both up- and down-satisfied, $\mathbf{b}$ will hold the solution to the monotone increasing least squares regression problem. This result is due to Kruskal.[22] Note that by convention the last block is automatically up-satisfied and the first block automatically down-satisfied. Monotone *decreasing* regression can be performed in a similar fashion.

 

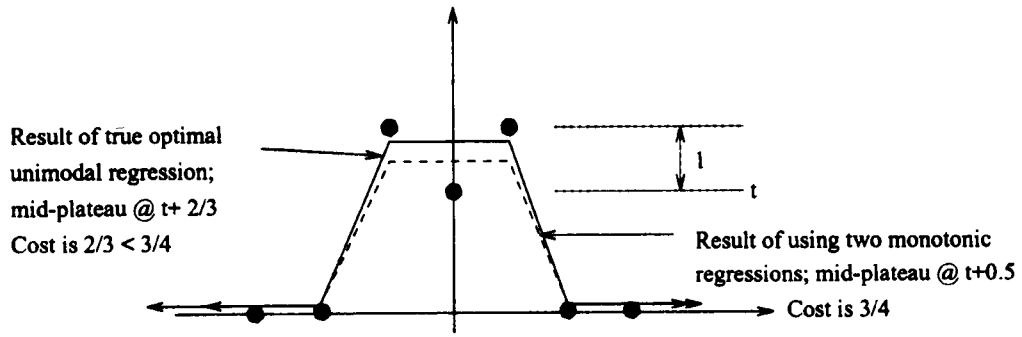


Figure 1. Two monotone regressions each including the hypothesized mode location are not equivalent to a unimodal regression. Regression input points are depicted as full hexagons. The difference in cost between the two solutions can of course be made arbitrary.

For unimodal regression with fixed mode location $n$, $1 \leq n \leq J$, we seek a vector $\mathbf{b}$ with a 'left' part that is monotone increasing, a right part that is monotone decreasing and a middle part that holds the maximum value:

$$\text{minimize} ||\boldsymbol{\beta} - \mathbf{b}||_F^2$$

$$\text{subject to}$$

$$b_1 \geq 0$$

$$b_j \geq b_{j-1}, \quad j = 2, \ldots, n$$

$$b_J \geq 0$$

$$b_j \leq b_{j-1}, \quad j = n+1, \ldots, J = \text{size}(\mathbf{b})$$

Here $n$ is given and not subject to optimization. The first important observation is that this problem is actually *not* equivalent to two monotone regression sub-problems, each involving location $n$, even if the two values assigned to location $n$ by the two respective monotone sub-regressions turn out to be identical. This is shown by means of a counter-example in Figure 1. The reason is that the two 'legs' of the regression are subject to coupling (interaction), albeit a loose type of interaction.

Suppose a vector $\mathbf{b}^L$ of size $(n-1) \times 1$ is the solution to the monotone increasing regression on the part of $\boldsymbol{\beta}$ to the left of the mode location. Similarly define $\mathbf{b}^R$ as the monotone decreasing regression on the right part of $\boldsymbol{\beta}$. Define the first interim candidate solution to our problem as

$$\mathbf{c} = \begin{bmatrix} \mathbf{b}^L \\ \beta_n \\ \mathbf{b}^R \end{bmatrix}$$

Note that the element $\beta_n$ is the $n$th element of the unconstrained least squares solution $\boldsymbol{\beta}$, $n$ being the mode location, i.e. $c_n = \beta_n$. Two situations might occur.

(*a*) $$c_{n-1} \leq c_n \geq c_{n+1}$$

In this case it follows immediately that the solution to the problem is the vector $\mathbf{c}$.

(*b*) $$c_n < c_{n-1} \text{ and/or } c_n < c_{n+1}$$

In this case $\mathbf{c}$ is not the solution, as the maximum is at $c_{n-1}$ (or $c_{n+1}$). In the following all averaging is performed over blocks, i.e. if $c_{n-1}$ is part of a block arising from the monotone regression from

 

which $c_{n-1}$ was computed, this will also be respected in the subsequent computations. Let $c_n$ be the active block. Find the neighboring block with the highest value. Concatenate and average over this block and the active block to get the new active block. Repeat the last two steps until no constraints are violated—the outcome will be the solution to our problem.

We call the above algorithm *ulsrfix* for unimodal least squares regression with fixed mode location. Although this is a well-known algorithm,[24] we include an alternative proof in the Appendix for two reasons: first, the particular method of proof is needed to prove an important result in the sequel; second, the method of proof is interesting in its own right.

One may use this fast algorithm for fixed mode location **ULSR** in conjunction with exhaustive search over all $J$ = size(**b**) possible mode locations to come up with an algorithm for **ULSR**. This is exactly what has been suggested in the past.[24,25] However, the exhaustive search over all $J$ possible mode locations is still quite annoying. This is addressed in the sequel by proposing an algorithm *ulsr* for solving problem **ULSR** which is then proven to be correct.

Suppose a monotone increasing regression is performed on $\boldsymbol{\beta}$. While calculating this regression vector, one also gets all the monotone increasing regressions for $\boldsymbol{\beta}^{1-j}$, $j = 1, \ldots, J$, $\boldsymbol{\beta}^{1-j}$ being a vector containing the first $j$ elements of $\boldsymbol{\beta}$. This can be derived as a side-benefit of Kruskal's monotone regression algorithm. Similarly a monotone decreasing regression for $\boldsymbol{\beta}$ will produce all monotone decreasing regressions for $\boldsymbol{\beta}^{1-j}$. The algorithm *ulsr* now proceeds as follows.

1. Calculate $\mathbf{b}^{I}$ as the monotone increasing regression on $\boldsymbol{\beta}$ and calculate $\mathbf{b}^{D}$ as the monotone decreasing regression on $\boldsymbol{\beta}$. Let $\mathbf{b}^{I,n}$ be the monotone increasing regression on the first $n-1$ elements of $\boldsymbol{\beta}$ and let $\mathbf{b}^{D,n}$ be the monotone decreasing regression on the last $J-n$ elements of $\boldsymbol{\beta}$.
2. For all $n$, $n = 1, \ldots, J$, define
$$\mathbf{c}^{(n)} = \begin{bmatrix} \mathbf{b}^{I,n} \\ \beta_n \\ \mathbf{b}^{D,n} \end{bmatrix}$$
3. Among only those $\mathbf{c}^{(n)}$ satisfying $\beta_n = \max(\mathbf{c}^{(n)})$, select a vector $\mathbf{c}^{(n)}$, which minimizes $||\boldsymbol{\beta} - \mathbf{c}^{(n)}||_F^2$, i.e.
$$\mathbf{b} = \underset{\mathbf{c}^{(n)} | \max(\mathbf{c}^{(n)}) = \beta_n}{\operatorname{argmin}} ||\boldsymbol{\beta} - \mathbf{c}^{(n)}||_F^2$$

As will be shown, this algorithm does provide a solution to problem **ULSR** and hence the above is algorithm *ulsr* for solving our stated problem. Note that in step 3 the candidate vectors $\mathbf{c}^{(n)}$ are only those for which $\beta_n = \max(\mathbf{c}^{(n)})$. These correspond exactly to those mode locations for which *ulsrfix* converges by two monotone regressions, without further averaging. Note also that even though the answer to the fixed mode location unimodal regression problem is unique (for it will be shown that it is a quadratic programming problem, so the fact that its solution is unique follows from uniqueness of the solution to a QP problem), the same is not necessarily true for unimodal regression.

### Claim 1

The unimodal regression problem, in which the mode location itself is not fixed but rather subject to optimization, always has a solution, but not necessarily a *unique* solution.

*Proof*

Consider the vector $[1\ 0\ 0\ 0\ 1]^T$. A unimodal regression is $[1\ 0.25\ 0.25\ 0.25\ 0.25]^T$, but so is $[0.25\ 0.25\ 0.25\ 0.25\ 1]^T$. ■

 

The following is an important result which proves correctness of algorithm *ulsr*.

**Theorem 1**

Consider the unimodal regression problem in which the mode location itself is subject to optimization and not fixed. Then, to optimize over all possible mode locations, one only needs to consider those candidate mode locations for which *ulsrfix* terminates in its first step; these probably contain the best mode location(s). What is more, as we have seen here, all required first steps of *ulsrfix* can be implemented simultaneously in just two full-size $J$ = size(**b**) Kruskal monotone regression steps.

*Proof*

See Appendix.

The importance of this theorem is twofold. First, it *proves* that one only has to consider a few candidate mode locations. Second, an exhaustive search using *ulsrfix* over all $J$ possible mode locations is certainly not necessary. Two modified Kruskal monotone regressions are sufficient.

The worst-case computational complexity of algorithm *ulsr* is upper bounded by $O(J^3)$ or $O(J^2)$, depending on whether or not one recomputes interim averages for improved numerical accuracy. As for Kruskal's regression, these upper bounds are most often overly pessimistic.

Comparing the complexity of algorithm *ulsr* with the algorithm suggested in Reference 24, it is readily seen that our algorithm is an order of magnitude faster. This is very important for the use of unimodality in multi-way algorithms, where *ulsr* is sometimes called thousands of times and its complexity constitutes the computational bottleneck.

One aspect not yet covered is how to implement non-negativity, but it follows immediately from Lemma 1 and the proof of Kruskal's monotone regression that one can simply set all negative values in the regression vector to zero. This will automatically lead to the optimal solution under the restriction of non-negativity.

In certain cases one is interested in more complicated constraints than mere unimodality and non-negativity. In the following we will discuss weighted unimodal least squares regression, equality-constrained unimodal least squares regression and oligomodal least squares regression. For further modifications, e.g. robust uni- and oligomodal regression, see Reference 26.

### 3.1. Weighted unimodal least squares regression

If information is available regarding the relative uncertainties of the elements on which the unimodal regression is performed, or if an iteratively reweighted approach is desired, algorithm *ulsr* can be modified to handle this. The problem to solve is the minimization of

$$||\mathbf{W} \circ (\mathbf{Y} - \mathbf{x}\mathbf{b}^{\mathrm{T}})||_{\mathrm{F}}^2$$

subject to **b** being unimodal, where **W** is a matrix of the same size as **Y** containing in its $ij$th element the uncertainty of the $ij$th element of **Y**. The operator $\circ$ is the Hadamard (element-wise) product. No closed-form solution seems possible in this case. Kiers,[27] however, has shown how to modify any unweighted ordinary least squares algorithm to a weighted algorithm. The basic principle is to iteratively fit the ordinary least squares algorithm to a transformed version of the data. This transformation is based on the actual data as well as on the current model estimate and the weights. In each step, ordinary least squares fitting to the transformed data actually improves the weighted least squares fit of the actual data. For details on this approach the reader is referred to Reference 27.

 

### 3.2. Equality-constrained unimodal least squares regression

Suppose a target is given that the vector being estimated should resemble or possibly equal. Such a situation can occur if for example the spectra of some analytes are known beforehand, but it can also occur in situations where the vectors being estimated are subject to equality constraints. Consider a situation where a set of vectors is to be estimated under unimodality constraints as in problem **UNIMODAL**. In some cases (see Section 4) it is known that a weighted sum of the vectors is equal to a constant vector, typically a vector of zeros or ones. As a simple example, if the matrix to be estimated, $\mathbf{B}$, consists of two unimodal column vectors that should be equal, this can be expressed algebraically as the equality constraint

$$\mathbf{CB} = \mathbf{d}$$

where

$$\mathbf{C} = [1 \quad -1], \qquad \mathbf{d} = [0]$$

General methods have been developed for solving linear problems under equality constraints, but a simple closed-form solution is not always possible owing to the unimodality constraints. Instead we can formulate the equality- and unimodality-constrained problem column-wise and use an iterative algorithm as described earlier. For a column of $\mathbf{B}$, say the first, $\mathbf{b}_1$, being estimated, the equality constraint can be expressed in terms of the current estimate of the other vector $\mathbf{b}_2$. The goal of the unimodality-constrained problem for a single vector is to find

$$\min_{\mathbf{b}} ||\mathbf{Y} - \mathbf{x}\mathbf{b}^{\mathrm{T}}||_{\mathrm{F}}^2 \tag{6}$$

subject to $\mathbf{b}$ being unimodal. For the equality-constrained problem we may consider the 'soft' constraint formulation

$$\min_{\mathbf{b}}(||\mathbf{Y} - \mathbf{x}\mathbf{b}^{\mathrm{T}}||_{\mathrm{F}}^2 + \lambda||\mathbf{g} - \mathbf{b}||_{\mathrm{F}}^2) \tag{7}$$

Here we have replaced $\mathbf{b}_1$ with $\mathbf{b}$ and $\mathbf{b}_2$ with the vector $\mathbf{g}$ which serves as the goal of the equality constraint. For other types of specific constraints, $\mathbf{g}$ may be defined accordingly. $\lambda$ controls the penalty levied for deviation from $\mathbf{g}$ (note that $\mathbf{g}$ is not the desired goal of the total problem). A very low value of $\lambda$ means that the solution $\mathbf{b}$ may deviate considerably from $\mathbf{g}$. A very large value of $\lambda$ would mean that the constraint should be essentially exactly fulfilled. In some cases it is desirable to use only a modest penalty in case one is not sure to what degree the applied constraint is appropriate, but mostly one is interested in estimating the solution under exact equality. For a given value of $\lambda$ the solution of the above hybrid problem may be obtained as (here $\boldsymbol{\beta}$ is exactly the same as in Lemma 1)

$$\begin{aligned}
\min(||\mathbf{Y} - \mathbf{x}\mathbf{b}^{\mathrm{T}}||_{\mathrm{F}}^2 + \lambda||\mathbf{g} - \mathbf{b}||_{\mathrm{F}}^2) & \\
&= \min(||\boldsymbol{\beta} - \mathbf{b}||_{\mathrm{F}}^2 + \lambda||\mathbf{g} - \mathbf{b}||_{\mathrm{F}}^2) \\
&= \min\left(\frac{1}{2}\mathbf{b}^{\mathrm{T}}\mathbf{b} - \boldsymbol{\beta}^{\mathrm{T}}\mathbf{b} + \frac{1}{2}\lambda\mathbf{b}^{\mathrm{T}}\mathbf{b} - \lambda\mathbf{g}^{\mathrm{T}}\mathbf{b}\right) \\
&= \min\left[\frac{1}{2}(1+\lambda)\mathbf{b}^{\mathrm{T}}\mathbf{b} - (\boldsymbol{\beta}^{\mathrm{T}} + \lambda\mathbf{g}^{\mathrm{T}})\mathbf{b}\right] \\
&= \min\left\{\frac{1}{2}\mathbf{b}^{\mathrm{T}}\mathbf{b} - [1/(1+\lambda)](\boldsymbol{\beta}^{\mathrm{T}} + \lambda\mathbf{g}^{\mathrm{T}})\mathbf{b}\right\} \\
&= \min[||1/(1+\lambda)(\boldsymbol{\beta} + \lambda\mathbf{g}) - \mathbf{b}||_{\mathrm{F}}^2] \\
&= \min||\mathbf{p} - \mathbf{b}||_{\mathrm{F}}^2
\end{aligned} \tag{8}$$

 

where

$$\mathbf{p} = 1/(1+\lambda)(\boldsymbol{\beta} + \lambda \mathbf{g}) \tag{9}$$

As can be seen from the above, the same algorithm can be used for solving the equality-constrained problem as for solving problem **ULSR**, by simply exchanging $\boldsymbol{\beta}$ with $\mathbf{p}$. Note that with this approach it is possible to impose approximate unimodality by exchanging $\mathbf{g}$ with the least squares unimodal solution and calculating the unconstrained solution to equation (8) which is equal to $\mathbf{p}$.

### 3.3 Oligomodal least squares regression

When bi- or oligomodal regression is sought, the approach developed so far is of little use. We will shortly outline a general algorithm for oligomodality by redeveloping the algorithm for unimodal least squares regression using a computationally more costly but also more flexible approach. The implementation details and/or modifications for a variation of the problem in hand are left to the reader.*

Consider again the **ULSR** problem with fixed mode location $n$. This **ULSR** problem can be cast as a special case of a standard quadratic programming (QP) problem.[28,29] In particular, observe that minimizing $||\boldsymbol{\beta} - \mathbf{b}||_2^2$ is equivalent to minimizing $\frac{1}{2}\mathbf{b}^T\mathbf{b} - \boldsymbol{\beta}^T\mathbf{b}$. In addition, for fixed $n$ the unimodality constraints can be put in matrix form, $\mathbf{Ab} \leq \mathbf{0}$, by defininining $\mathbf{A}$ appropriately. As a result, for fixed mode location this **ULSR** problem can be cast as

$$\begin{aligned} &\text{minimize } \frac{1}{2}\mathbf{b}^T\mathbf{b} - \boldsymbol{\beta}^T\mathbf{b} \\ &\text{subject to } \mathbf{Ab} \leq 0 \end{aligned}$$

which is a special case of a standard form of quadratic program.[28,29]

That said, one may solve Problem 1 itself by simply trying all $J$ = size($\mathbf{b}$) possible mode locations and selecting the one that gives minimum error. This exhaustive search is certainly neither computationally appealing nor conceptually elegant. In particular, its worst-case complexity is $O(J^5)$. We shall describe an interesting alternative. Let us slightly change the **ULSR** problem definition. In particular, let us further constrain the individual elements of the solution vector $\mathbf{b}$ to take on values in a finite collection ('alphabet') of values. In effect we consider a discrete or quantized version of the **ULSR** problem. One may just as well think of it as a further restricted **ULSR** problem. Thus the optimal solution to the discrete problem is never better than the optimal solution to the original **ULSR** problem.

The motivation for doing so is twofold. First, if one considers sufficiently many and properly chosen discrete levels, then one may approximate the original **ULSR** problem and therefore its solution as finely as one wishes. Second, the discrete problem admits a fast and elegant one-step solution which jointly optimizes *both* mode location *and* output vector levels. Let us now formally state the discrete (non-negative) unimodal least squares regression (**DULSR**) problem.

**Problem 2**

**(DULSR)** Given a $J \times 1$ vector $\boldsymbol{\beta}$,

$$\begin{aligned} &\text{minimize} ||\boldsymbol{\beta} - \mathbf{b}||_F^2 \\ &\text{subject to } \mathbf{b} \in \mathscr{F}_J(\mathscr{E}) \end{aligned}$$

* An algorithm for problem **ULSR** (as well as other problems) based on dynamic programming is available from the second author on request.

 

where $\mathcal{F}_J(\mathcal{E})$ is the set of all unimodal vectors of $J$ elements of $\mathcal{E}$, $|\mathcal{E}| < \infty$, i.e. has only a finite number of elements. Let $R = |\mathcal{E}|$, i.e. the number of discrete values in the set $\mathcal{E}$. Consider a graph consisting of $J$ stages (sets of nodes), each stage consisting of $2R$ nodes. In this context the $j$th stage refers to the $j$th element of **b**. The nodes of each stage are arranged in a vertical fashion and subsequent stages are laid out parallel to each other to the right of stage one. Each node holds three node variables: a cost variable, a flag variable and a pointer variable. The $2R$ nodes of any given stage are partitioned into two subsets of $R$ nodes each. Each subset contains exactly one node for each element of $\mathcal{E}$; thus we tag each node with its corresponding element of $\mathcal{E}$. The nodes of the first subset have their associated flag variables set to zero. The nodes of the second subset have their associated flag variables set to one. Thus at each stage for any given node there exists a companion node having the same tag but a different flag variable. The flag variable indicates whether the current variable is to the left or right of the position of the maximum, with the flag one also used at the position of the maximum. Initially, subsequent stages are fully connected in the sense that it is possible to go from any node at stage $j$ to any node at stage $j + 1$. Let us visualize these connections by means of directed arcs emanating from any node at stage $j$ and pointing to all the nodes at stage $j + 1$.

Now suppose we selectively prune certain arcs. In particular, suppose we remove all arcs emanating from a node whose associated flag variable is one and pointing to a node whose associated flag variable is zero. In addition, we remove any arc in between an origin and a destination node if either node has its associated flag equal to zero and the tag of the origin is strictly greater than the tag of the destination; similarly, we remove any arc in between an origin and a destination if the origin flag is equal to one and the tag of the origin is strictly less than the tag of the destination. Next consider all the remaining paths in the resulting directed graph which start from a node at the first stage and terminate in a node at the $J$th stage. The claim is that the collection of all these paths can be identified with $\mathcal{F}_J(\mathcal{E})$, the set of unimodal vectors consisting of $J$ elements of $\mathcal{E}$. One may easily verify that, starting from any node at the first stage, one may only traverse paths along the graph that correspond to unimodal sequences of associated tags*. Indeed, once a decreasing tag transition is made, one is forced to follow paths through nodes whose associated flag is one; from these, only non-increasing tag transitions are possible.

Now the problem is amenable to a dynamic programming (DP) solution.[30–33] DP may be thought of as a shortest path algorithm on an ordered directed graph. The only thing that remains is to specify suitable costs associated with the remaining arcs (allowable transitions) in the graph: for each arc pointing to, say, a node at stage $j$ the associated cost is the squared Euclidean distance between the tag of the node to which the arc points and $\beta_j$. DP proceeds recursively as follows. First all the nodes at stage one are visited in turn, their cost is computed as the squared Euclidean distance between the tag of each node and $\beta_1$ and the result is stored in the respective node cost variables. Then all the nodes at stage two are visited in turn. For each node one looks back at all the nodes at the previous stage that have access to the node in hand and finds the one whose cost is minimal; once this best 'predecessor' is found, one adds to this minimal cost the squared Euclidean distance between the tag of the node in hand and $\beta_2$, stores the result in the node-in-hand cost variable, updates the node-in-hand pointer variable to point to this best predecessor, then proceeds to the next node, then the next stage, and so on and so forth.

In the end, once one reaches the $J$th stage, one simply selects the minimum cost path over all the paths terminating at stage-$J$ nodes and traces this path backwards using the node pointer variables.

* More than one such path may correspond to the same element of $\mathcal{F}_J(\mathcal{E})$. This happens because the location of the maximum need not be uniquely defined for certain elements of $\mathcal{F}_J(\mathcal{E})$, namely those with flat peak plateaux. This non-uniqueness is not a concern, for all these paths have the same cost.

 

During this traceback the optimal digital unimodal regression sequence is output (in reverse order) by printing the tags of nodes visited.

Optimality of this procedure can be shown mathematically by mimicking the development in Reference 34 (see also Reference 35). However, the somewhat informal development presented here is more intuitive than the formal mathematical argument. What makes things work is the so-called *principle of optimality* of DP.[30,31]

One may easily verify that the complexity of this program is $O(R^2J)$, where again $J$ is the length of the regression input (and output) vector and $R = |\mathscr{E}|$. Note that, for example, equality constraints can easily be incorporated in this framework and actually help further lower computational complexity.

A simple alternative method for solving the **ULSR** problem then naturally suggests itself. The mode location is determined by **DULSR**, then, given the mode location so determined, one may use either a QP-based algorithm or, better, our own *ulsrfix* to find the solution. Note that this approach is more complex than our previous proposal. It is also suboptimal in the sense that the mode localization step via **DULSR**, even though optimal in the discrete setting, is suboptimal in terms of the original **ULSR** problem owing to its finite resolution. On the other hand, for an adequate number of discrete levels this is a small problem and the benefit of this alternative approach is that both DP and QP easily allow one to incorporate further constraints, e.g. equality and/or inequality constraints. In addition, this alternative approach may handle constraints such as oligo modality in a straightforward manner. Oligomodality may not easily be addressed by our first proposed algorithm.

## 4. EXPERIMENTAL

To test the new algorithm, several examples of its use will be shown. We will not give any examples arising from chromatography or electrophoresis. The reason is that the problems in which we are currently interested do not stem from these areas, but also that unimodality as a concept has been utilized for a long period in problems from these areas. Hence the usefulness there is almost self-evident. For demonstration purposes a simple simulated problem will first be presented to illustrate what unimodality does. Then an example arising from spectrofluorometry and an example from flow injection analysis will be described.

### 4.1. Simulated example

A matrix $\mathbf{X}$ is constructed as

$$\mathbf{X} = \mathbf{a}\mathbf{b}^{\mathrm{T}} + \mathbf{E} \tag{10}$$

where $\mathbf{a}$ is a $25 \times 1$ vector of uniformly distributed random numbers and $\mathbf{b}$ is a $100 \times 1$ vector with a Gaussian shape. The matrix $\mathbf{E}$ consists of normally distributed random numbers with a standard deviation varying from 1% to 250% of the maximal value of the systematic data $\mathbf{a}\mathbf{b}^{\mathrm{T}}$. A Gaussian shape was used in this example for practical reasons. Even though constraining the estimated $\mathbf{b}$ vector to be Gaussian in this case would be appropriate, in the real-world examples given in the sequel the profiles to be estimated are not at all Gaussian, though still unimodal.

In Figure 2 the results of estimating a bilinear ALS model are shown for different noise levels. In each case the unconstrained, non-negativity-constrained and unimodality-constrained solutions are shown for the estimated $\mathbf{b}$. The true $\mathbf{b}$ is also shown. It is easily seen that as the noise increases, the structure of the unconstrained and non-negativity-constrained solutions vanishes, while the unimodality-constrained solution still resembles the true $\mathbf{b}$. One may also note that for very high noise levels the unimodality-constrained profile tends to get 'spiky'. This is not an artifact but due to the definition of unimodality. In cases with very high noise levels where such spikes are observed, one may want to further constrain the solution to obey not only the unimodality constraints but also a

 

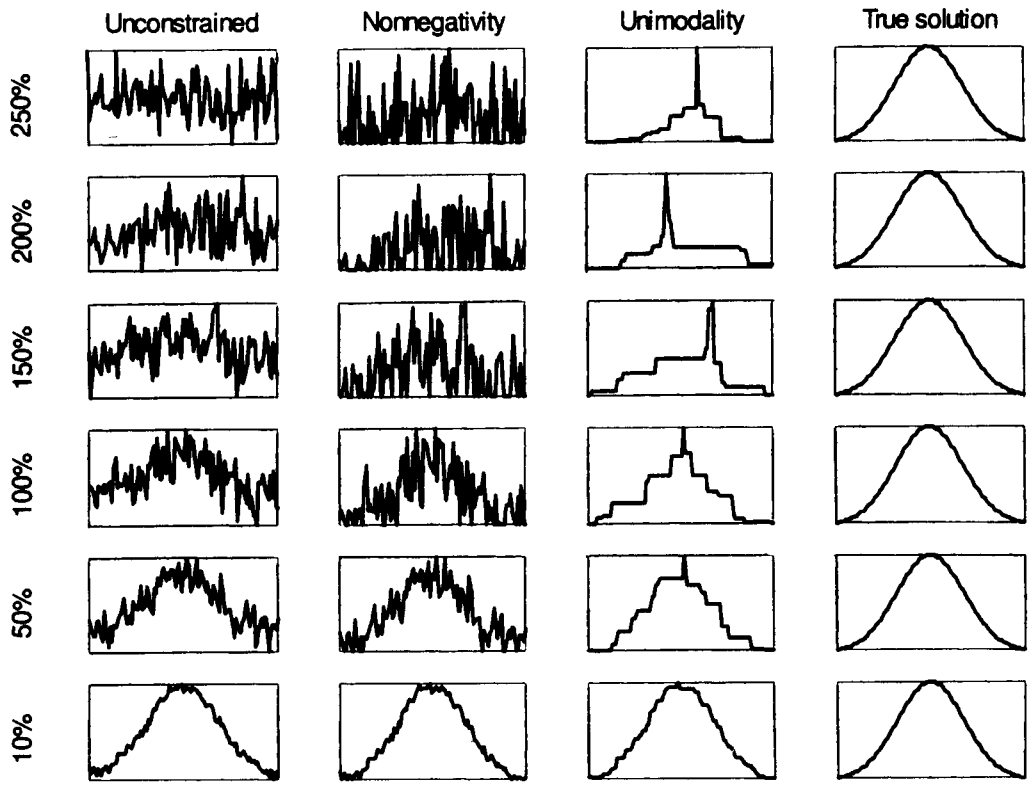


Figure 2. Result of estimating **b** using different levels of noise (noise level shown to left). Arbitrary units. See text for further explanation.

smoothness constraint. Several ways of implementing such a constraint could be envisioned but will not be pursued here.

Applying appropriate constraints as here can be helpful not only in estimating the specific parameters pertaining to the constraints but also to get a more valid model overall. As an example, Table 1 lists the correlations between the true **a** and the estimated **a** for all the models shown in Figure 2.

As can be readily seen, the unimodality-constrained model is consistently better in estimating the profile **a**. This improvement is more pronounced the higher the noise level, i.e. the more difficult it is to model the data, the more useful appropriate constraints are. In this case the data to be modeled were very simple in that the only deviation from the true model was random noise. For real data many other types of deviations are likely to occur, making the usefulness of constraints even more pronounced. Next we demonstrate the usefulness of unimodal regression on real data.

### 4.2. Fluorescence spectroscopy

Sugar was sampled every eighth hour during a campaign from a sugar plant in Scandinavia giving a total of 268 samples (approximately 3 months), of which three were discarded in this investigation. Each sugar sample was dissolved in water, 2.25 g/15 ml, and the solution was measured spectro-

 

Table 1. Quantitative results from analysis of different noise levels

| | Correlation between estimated and true profile **a** | | |
|---|---|---|---|
| Noise level (%) | Unconstrained | Non-negativity | Unimodality |
| 250 | 0·06 | 0·45 | 0·54 |
| 200 | 0·32 | 0·43 | 0·76 |
| 150 | 0·62 | 0·64 | 0·69 |
| 100 | 0·93 | 0·93 | 0·94 |
| 50 | 0·96 | 0·96 | 0·97 |
| 10 | 1 | 1 | 1 |

fluorometrically in a cuvette in a PE LS50B spectrofluorometer. Raw non-smoothed data were used. For every sample the emission spectrum from 275 to 560 nm was measured in 0.5 nm intervals (571 wavelengths) at seven excitation wavelengths (230, 240, 255, 290, 305, 325, 340 nm). The data can consequently be arranged in an $I \times J \times K$ three-way array of specific size $265 \times 571 \times 7$. The first mode refers to samples, the second to emission wavelengths and the third to excitation wavelengths. The $ijk$th element in this array thus corresponds to the measured emission intensity from sample $i$ excited at wavelength $k$ and measured at wavelength $j$. For weak solutions, fluorometric data can theoretically be described by a PARAFAC model, with the exception that for each sample the measured excitation–emission matrix (size $J \times K$, specifically $571 \times 7$) has a part that is systematically missing in the context of the trilinear model.[36] Very crudely, one can say that emission is not defined below the wavelength at which the sample is excited. In practice, owing to Rayleigh scattering, one will also find that emission slightly above the excitation wavelength does not conform to the trilinear PARAFAC model. As the PARAFAC model only handles regular three-way data, one needs to set the elements corresponding to non-trilinear areas to missing, so that the estimated model is not skewed by these data points. In this case the implication of this is that a rather large part of the data is missing in the emission area from 260 to 340 nm, hence making the profiles prone to some instability in this area.

As has been described earlier, the PARAFAC model is intrinsically unique under mild conditions and the hope with these data is that one should be able to estimate components that are chemically meaningful and hence provide a direct connection between the production (the sugar samples) and the chemical understanding of quality. In this paper it is sufficient to state that we would like to have components that are plausible and can be related to the chemistry of sugar production. When estimating a four-component PARAFAC model with non-negativity constraints, four pseudo-concentration profiles are obtained, each with corresponding pseudo-excitation and emission spectra. The components are pseudo-spectra and concentration profiles in the sense that they are estimated from the composite fluorescence data, but may be estimates of real analytes. This can be verified much like analytes are identified in chromatography, but it is not of primary concern here.

The model is first estimated under non-negativity constraints, as spectral parameters as well as concentrations are known to be non-negative. For identification purposes the estimated excitation spectra are not optimal as they are only of dimension $7 \times 1$ and hence difficult to assess and discern. The estimated emission spectra, on the other hand, are very amenable to qualitative and quantitative assessment. In Figure 3(a) the estimated emission spectra are shown. From visual inspection the spectra seem mainly reasonable, but for one spectrum the bump slightly above 300 nm seems to be more of a numerical artifact than real.

To possibly substantiate this visual judgement, a split-half experiment was performed.[37] A split-half experiment is a type of bootstrap analysis where specific subsets are analysed independently. In

 

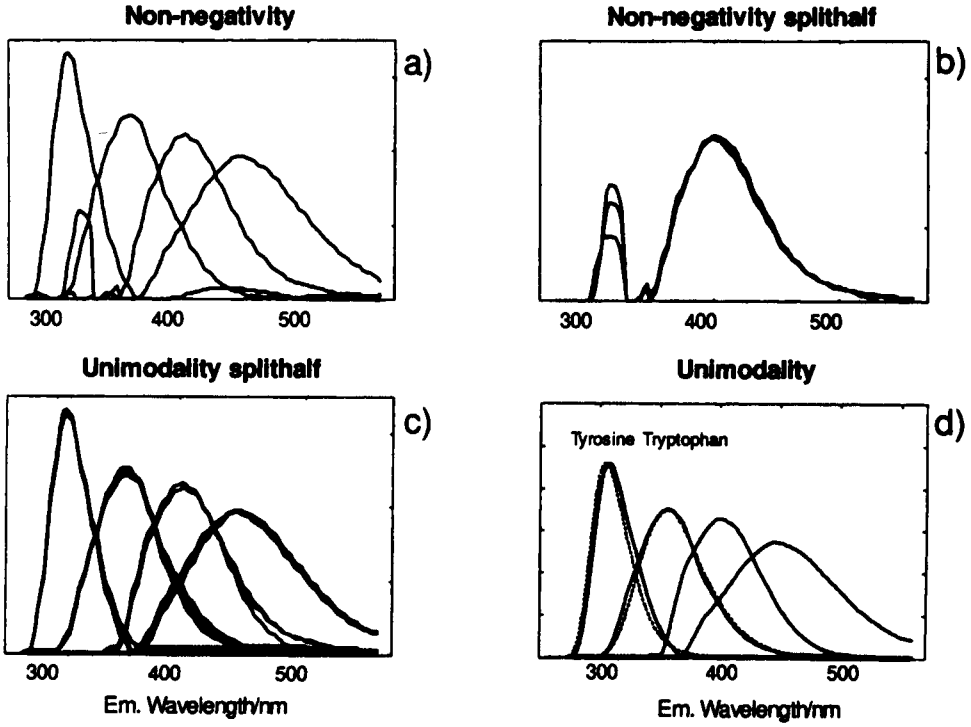


Figure 3. Estimated emission spectra from fluorescence data: (a) four spectra estimated using non-negativity; (b) suspicious spectrum estimated from different subsets using non-negativity; (c) estimated spectra from different subsets using unimodality; (d) comparing estimated spectra with spectra of tyrosine and tryptophan (shown with dotted lines). Arbitrary ordinate units.

this case, using different sets of samples for estimating the four-component model should give essentially the same estimated emission spectra, as the uniqueness of the model should not depend on the specific set of samples used as long as the samples cover the same population. The samples were divided into two contiguous sets (A and B) of approximately the same size. As the two sets cover different time spans, they can be considered to be completely independent samplings. Thus, if the same emission spectra appear in submodels of these data sets, it will be a strong indication that the model is reflecting real underlying phenomena rather than phenomena generated by noise or sampling variation. However, it may happen that in one of the sets some phenomena do not appear because they are simply absent in the corresponding period. To prevent this, two other data sets were generated, C and D. The set C consists of the first half of samples from sets A and B and the set D consists of the last half of samples from sets A and B. These four sets (A, B, C, D) are pairwise completely independent. Similarity of the estimated spectra in either set A *and* set B or in set C *and* set D will be evidence of real phenomena, but of course similarity between all model estimates is hoped for. The resulting model estimates of the problematic emission spectrum are shown in Figure 3(b). One estimate is left out as the corresponding model did not converge to a meaningful solution. The area around 300 nm is seen to be unstable in a split-half sense. The estimated parameters in this region change depending on which subset of samples is used for estimating the model, whereas the remaining parameters are more or less insensitive to subset variations. The split-half experiment thus confirms that the area is ill-modeled. The following features all indicate that the estimated area is unreliable.

 

(a) The parameters are visually off-the-mark in the sense that wavelength-to-wavelength changes are not smooth.
(b) The split-half experiment shows that the parameters cannot be identified in a stable fashion.
(c) The fact that the data contain many missing values in the area of the unstable region explains why the instability occurs.

The question then is what to do. As the most probable cause of the problem is that too few excitation wavelengths have been used (seven), the best thing to do would probably be to remeasure the samples using more excitation wavelengths. However, the measurements as they are currently being performed require substantial work, so remeasuring is not realistic. For future samples more excitation wavelengths may be used, but for these data the only possibility is to remedy the artifact by means of data processing. Several aspects indicate that the spectrum should really be unimodal.

(a) The spectrum *is* unimodal apart from the unstable part.
(b) The remaining estimated emission spectra are almost unimodal.
(c) The most likely fluorophores in sugar (amino acids and derivates) have unimodal emission spectra.
(d) The Kasha rule[38] states that a fluorophore will emit light under the same ($S_1$–$S_0$) transition regardless of excitation, i.e. an excited molecule will drop to the lowest vibrational level through radiationless energy transfer and then from the excited singlet level $S_1$ return to the ground state $S_0$ by fluorescence.[36] Even though there are exceptions to this rule, it often holds, especially for simple molecules. The fact that the emission occurs from the same transition mostly implies that the corresponding emission spectrum will be unimodal.

The above reasoning led to specifying a new model where all emission spectra were estimated under unimodality constraints and remaining parameters under non-negativity constraints. The estimated model was stable in a split-half sense (Figure 3(c)) and interestingly the estimated excitation spectra and relative concentrations did not change much from those of the non-negativity-constrained model. This verifies that the artifact is mainly due to the amount of missing data in the specific region. The estimated emission spectra are shown in Figure 3(d) together with the emission spectra of tyrosine and tryptophan, two substances of known technological importance. These spectra were acquired in experiments unrelated to this study. Nevertheless, the similarity confirms that the PARAFAC model is capturing chemical information and hence provides means to relate technological aspects and detailed chemical understanding.

### 4.3 Flow injection analysis

In Reference 39 an analysis was performed on data arising from flow injection analysis (FIA). We will use part of these data to exemplify how equality and unimodality constraints can be useful. In FIA there is essentially no separation of the analytes and hence all analytes will have the same time profile. In this particular case, however, the samples contained different amounts of 2-hydroxy-benzaldehyde (2-HBA), 3-hydroxy-benzaldehyde (3-HBA) and 4-hydroxy-benzaldehyde (4-HBA), which all have different acidic and basic spectra. As a pH profile was induced over time, the analyte's acidic and basic forms are present at a specific time in different amounts depending on the analyte $pK_a$ value and the pH at that specific time. For a single-analyte sample a theoretical structural model of the measurements can be given as

$$\mathbf{X}_f = c_f \mathbf{s}_{af} \mathbf{p}_{af}^T + c_f \mathbf{s}_{bf} \mathbf{p}_{bf}^T = c_f (\mathbf{s}_{af} \mathbf{p}_{af}^T + \mathbf{s}_{bf} \mathbf{p}_{bf}^T) \tag{11}$$

where $\mathbf{s}_{af}$ is the spectrum of the $f$th analyte in its acidic form and $\mathbf{s}_{bf}$ is the spectrum of the basic form

 

of the analyte. The vector $\mathbf{p}_{af}$ is the time profile of the analyte in acidic form and $\mathbf{p}_{bf}$ of the analyte in basic form. The concentration of the analyte in the sample is $c_f$. Now let $\mathbf{S}_f = [\mathbf{s}_{af}\ \mathbf{s}_{bf}]$ and $\mathbf{P}_f = [\mathbf{p}_{af}\ \mathbf{p}_{bf}]$. Then the measured data can also be expressed as

$$\mathbf{X}_f = c_f \mathbf{S}_f \mathbf{P}_f^T \tag{12}$$

For a sample with several analytes the theoretical model becomes

$$\mathbf{X} = \sum_{f=1}^{F} \mathbf{X}_f = \sum_{f=1}^{F} c_f \mathbf{S}_f \mathbf{P}_f^T \tag{13}$$

Extending the model to several samples, the following general model is obtained:

$$\mathbf{X}_i = \sum_{f=1}^{F} \mathbf{X}_{if} = \sum_{f=1}^{F} c_{if} \mathbf{S}_f \mathbf{P}_f^T \tag{14}$$

where $\mathbf{X}_i$ is the measurement made on the $i$th sample, $\mathbf{X}_{if}$ is the contribution from the $f$th analyte to the $i$th sample and $c_{if}$ is the concentration of the $f$th analyte in the $i$th sample. As can be readily seen from the model, only the concentrations of the analytes change from sample to sample; the spectra and time profiles remain the same, as these are intrinsic parameters related to the chemical and physical system respectively. An ALS algorithm for estimating the above model in a least squares sense can be constructed by simply modifying a PARATUCK2 algorithm.[40]

A data set of twelve samples was modeled using the restricted PARATUCK2 model. In each sample the same three analytes were present but in different concentrations. First a model was estimated using only non-negativity and then a second model using additional equality and unimodality constraints for estimating the time profiles. To understand this very restricted model, consider the time profile in general of a FIA system. As a FIA system is only a transportation system, it does not separate analytes. Only dispersion occurs, and in this case, as the analytes are chemically very similar, the dispersion will be almost identical. Therefore all analytes will have the same time profile shape in a FIA system. In this particular FIA system a pH profile has been induced over time such that the pH of the fluid changes gradually from very high (11.4) at the beginning of the sample plug to very low (4.5) at the end of the sample plug. All three analytes will have identically shaped time profiles, but owing to the pH changes the profile will be the sum of a contribution from the basic analyte ($\mathbf{p}_{bf}$) and a contribution from the analyte in acidic form ($\mathbf{p}_{af}$). These profiles will differ for different analytes owing to the difference in p$K_a$. When estimating the time profiles, we will thus have an estimation problem where six profiles are to be estimated (three analytes each having an acidic and a basic profile), but the additional equality constraint requires that the sum of the two profiles for one specific analyte will have the same (unknown) shape for all analytes. Further, we require the profiles to be non-negative for obvious reasons and also to be unimodal. The unimodality of individual profiles follows from the unimodality of a dispersion profile in a reasonably well-behaved FIA system and the continuity of the pH profile.

The way we will compare the results of the two models is by showing

(i) how well the concentration profiles (three for each model) agree with the reference concentrations of the three analytes (known, as the samples are laboratory-made) and
(ii) how well the acidic and basic spectra are estimated. The spectra of the pure analytes are determined from standard two-way curve resolution techniques of measurements on pure samples. The corresponding spectra have been verified experimentally.

The agreement can be monitored by the correlation coefficients between estimates and reference

 

Table 2. Correlation between estimated and reference concentrations. 'All constraints' means non-negativity, unimodality and equality as described in text

| Constraints | 2-HBA | 3-HBA | 4-HBA |
|---|---|---|---|
| Non-negativity | 0·9988 | 0·9787 | 0·9996 |
| All constraints | 0·9992 | 0·9987 | 0·9996 |

Table 3. Correlation between estimated and reference spectra. 'All constraints' means non-negativity, unimodality and equality as described in text

| | 2-HBA | | 3-HBA | | 4-HBA | |
|---|---|---|---|---|---|---|
| Constraints | Acidic | Basic | Acidic | Basic | Acidic | Basic |
| Non-negativity | 0·9944 | 0·9117 | 0·9952 | 0·9241 | 0·9974 | 0·9977 |
| All constraints | 0·9946 | 0·9590 | 0·9953 | 0·9989 | 0·9966 | 0·9943 |

data. In Tables 2 and 3 these correlation coefficients are shown. As can be seen, just using non-negativity constraints, one obtains excellent results. Both concentrations and spectra are well estimated by the model. However, the interesting point here is that if additional constraints are known to be valid, using these as well will improve the model, as evidenced by the consistently better model obtained using all three mentioned constraints. All correlations are either identical or substantially improved. For illustrative purposes the basic spectra of 2-HBA estimates as well as the reference are shown in Figure 4.

## 5. CONCLUSIONS

A fast algorithm for estimating unimodal and non-negativity-constrained solutions for a well-known problem in many chemical applications has been developed and tested. The algorithm is robust and amenable to a wide variety of modifications, as shown here by using it in conjunction with equality

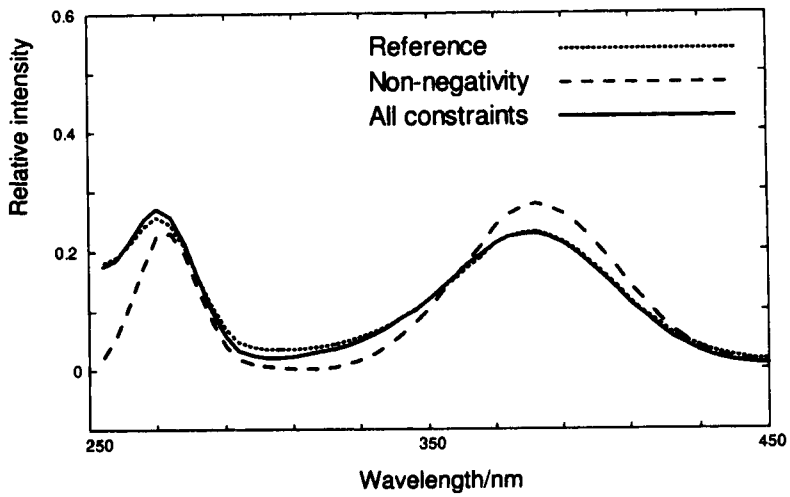


Figure 4. Basic spectra of 2-HBA: reference as well as estimated using non-negativity and using all constraints.

 

constraints and by refining the algorithm to a weighted least squares loss function. We have further described an algorithm for unimodality-constrained regression with fixed mode location and also outlined how to obtain an algorithm for the more general oligomodal regression problem.

## 6. MATERIALS AND METHODS

The algorithms have been implemented in MATLAB for Windows v5.0 (MathWorks, Inc.) and can be obtained from the Internet at http://newton.foodsci.kvl.dk/foodtech.html. All calculations were performed on a 200 MHz Pentium Dell PC with 64 Mb RAM. The data used are also available from the first author.

## APPENDIX

**Proof of Lemma 1**

Let

$$\boldsymbol{\beta} = \mathbf{x}^T\mathbf{Y}/\mathbf{x}^T\mathbf{x}$$

i.e. the solution to the unconstrained problem, and

$$\mathbf{E} = \mathbf{Y} - \mathbf{x}\boldsymbol{\beta}^T$$

Then

$$\min_{\mathbf{b}} ||\mathbf{Y} - \mathbf{x}\mathbf{b}^T||_2^2 = \min_{\mathbf{b}} ||\mathbf{x}\boldsymbol{\beta}^T + \mathbf{E} - \mathbf{x}\mathbf{b}^T||_F^2$$

$$= \min_{\mathbf{b}}[\mathrm{tr}(\mathbf{E}^T\mathbf{E}) + 2\mathrm{tr}\mathbf{E}^T\mathbf{x}(\boldsymbol{\beta} - \mathbf{b})^T + \mathrm{tr}(\boldsymbol{\beta} - \mathbf{b})\mathbf{x}^T\mathbf{x}(\boldsymbol{\beta} - \mathbf{b})]$$

As $\mathrm{tr}(\mathbf{E}^T\mathbf{E})$ and $\mathbf{x}^T\mathbf{x}$ are constant and $\mathrm{tr}(\mathbf{E}^T\mathbf{x})$ is a vector of zeros, it follows that

$$\min_{\mathbf{b}} ||\mathbf{Y} - \mathbf{x}\mathbf{b}^T||_F^2 = \min_{\mathbf{b}}[\mathrm{tr}(\boldsymbol{\beta} - \mathbf{b})^T(\boldsymbol{\beta} - \mathbf{b})] = \min_{\mathbf{b}} ||\boldsymbol{\beta} - \mathbf{b}||_F^2$$

Hence the **ULSR** Problem 1 above is equivalent to

$$\text{minimize} ||\boldsymbol{\beta} - \mathbf{b}||_F^2$$
$$\text{subject to } \mathbf{b}\text{: } \textit{unimodal}$$

■

**Proof of correctness of algorithm *ulsrfix***

The following lemmas and theorem prove that *ulsrfix* indeed produces the sought solution to the problem.

**Lemma 2**

The proposed *ulsrfix* algorithm terminates in *at most* $J$ = size(input) steps, with a feasible (i.e. unimodal) solution, for the given fixed mode location.

*Proof*

This is trivial, since the size of one of the two auxiliary monotone legs used by the algorithm in interim computations decreases by at least one in each iteration and the combined length of both legs is initially size(input) $-1$. The algorithm will go through all size(input) steps if interim steps do not

 

provide a unimodal solution; and if this happens, the final suggestion is flat (trivially unimodal). ■

**Claim 2**

Consider the fixed mode location unimodal regression problem and leave out some of the elemental constraints (e.g. $b_j \leq b_{j+1}$ if $j$ is to the left of the mode location). A true regression (optimal fit) over the remaining constraints will give a fit that is no worse than the fit of true fixed mode location unimodal regression. This is because the former is a less constrained problem.

**Lemma 3**

At each step, *ulsrfix* provides a true regression over a subset of the given fixed mode location unimodality constraints.

*Proof*

At the first step this is trivially true: the first interim solution is a regression over

$$b_1 \leq \ldots \leq b_{n-1}, \quad b_{n+1} \geq \ldots \geq b_J$$

where $n$ is the given mode location and $J = \text{size}(\mathbf{b})$.

Let $\boldsymbol{\beta}$ be the input. At the second step, if we average $\beta_n$ with the highest block *of the higher of the two monotone regression legs*, say, without loss of generality, the highest batch of the right leg, denote its leftmost and rightmost positions by $[l,r]$ and let its level before the new averaging be *MAX*, then, equivalently, we optimally enforce the constraints

$$MAX \geq b_n \geq b_l \geq \ldots \geq b_r, MAX \geq b_{r+1} \geq \ldots \geq b_J$$

Now the claim is that these constraints can be considered a subset of the original fixed mode location unimodality constraints. The argument goes as follows. First, the fact that optimizing over $b_n \geq b_l \geq \ldots \geq b_r$ is equivalent to optimizing over $b_n = b_l = \ldots = b_r$ is a consequence of correctness of Kruskal's monotone regression algorithm. In particular, since $[b_l \ \ldots \ b_r]$ is the *last block* of an increasing Kruskal regression, by the causal order in which Kruskal's algorithm processes the data, it follows that the location $r$ was down-satisfied (since it was not down-averaged) and thus the regression *decoupled* at $b_r$ and remained decoupled (since otherwise the block would have been down-averaged). It follows that $[b_l \ \ldots \ b_r]$ is a pure monotone regression over indices $[l,r]$. Since this regression led to a complete averaging and the value at location $n$ is below this average, it follows from correctness of Kruskal's algorithm that a regression over $b_n \geq b_l \geq \ldots \geq b_r$ is equivalent to regression over $b_n = b_l = \ldots = b_r$.

Second, unless *ulsrfix* terminates in its first step, no element of the true fixed mode location unimodal regression vector can ever be greater than this underconstrained *MAX* anyway; so $MAX \geq b_n$ and $MAX \geq b_{r+1}$ can be thought of as part of the original constraints and, for the data in hand, the problem is not altered in any way.

Of course, the lower monotone regression leg remains unaltered, thus still part of a true regression over a subset of the original constraints. Hence the new total interim solution is a true regression with respect to a subset of the original constraints. These arguments now carry over to subsequent steps and the proof of this lemma is complete.

Notice the following delicate point: if we instead average $\beta_n$ with the highest batch of the *lower* of the two monotone regression legs, then we are effectively introducing *a new and arbitrary tentative constraint*, namely that the remainder of this leg be less than or equal to the maximum of the full leg before averaging. This is arbitrary and voids the proof of the subsequent theorem. Therefore *we*

 

*should always average with the highest batch of the higher of the two legs.* ■

**Claim 3**

Thus at each step of *ulsrfix* the fit of the interim solution is no worse than that of the true fixed mode unimodal regression.

**Theorem 2**

The proposed *ulsrfix* algorithm is *correct*, i.e. it provides a true (optimal) regression with respect to the given fixed mode location unimodality constraints.

*Proof*

From the above lemmas and claims it follows that since interim solutions have a fit that is no worse than the fit of the true fixed mode location unimodal regression, and the algorithm terminates in a finite number of steps at a feasible (unimodal) solution, this unimodal solution must have a fit that is no worse than the fit of the true fixed mode location unimodal regression and therefore should be the true fixed mode location unimodal regression itself. ■

One may show that Kruskal's original monotone regression algorithm has an associated worst-case computational complexity which is (rather loosely) upper bounded by $O(J^3)$ or $O(J^2)$, where $J$ = size(input), depending on whether or not one recomputes interim averages for improved numerical accuracy. The actual run time of Kruskal's algorithm is usually sub-quadratic.

**Lemma 4**

The worst-case complexity of *ulsrfix* is exactly the same as the worst-case complexity of Kruskal's monotone regression algorithm.

*Proof*

The first step of *ulsrfix* entails computing two Kruskal regressions of combined size $J - 1$, where $J$ = size(**b**). Since Kruskal's algorithm is sub-linear, the worst case for this step is when only one monotone regression of size $J - 1$ is needed. Subsequent steps of *ulsrfix* consist of a constant number of operations each and there is a grand total of at most $J - 1$ of these steps. It follows that the complexity bottleneck is the first step of *ulsrfix* and the proof is complete. ■

**Proof of Theorem 1**

Consider Figure 5. It depicts the five possible configuration classes for the input value at a hypothesized mode location relative to the two monotone regression legs after the first step of *ulsrfix*.

Observe that any given unimodal solution may not have a unique maximum owing to the possibility of plateaux (flats) in the mode neighborhood. Notice that this is a non-uniqueness due to semantics, not related to the non-uniqueness of the problem of optimized mode location unimodal regression. In such cases any one of the indices of this maximal plateau may be equally well considered to be the mode location; constraining for mode location at either one of these indices will necessarily produce the same result by virtue of optimality. Hence we may, without loss of generality, restrict our search for an optimal mode location to a search for an optimal *leftmost* mode location; this does not sacrifice optimality in any way. Consider case (b) in Figure 5. Clearly the optimal fixed mode unimodal regression for the given mode location will entail a higher cost than the current

 

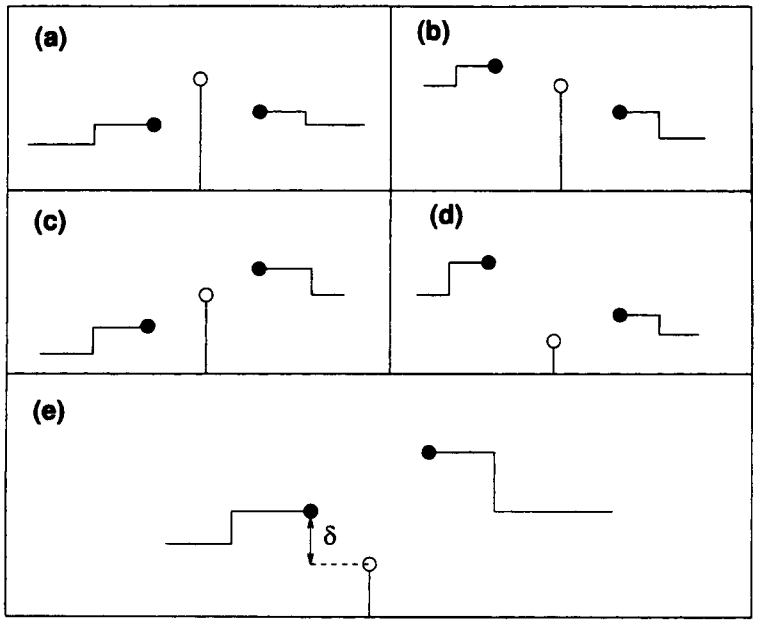


Figure 5. After the first step of *ulsrfix*, the five possible configuration classes for the input value at a hypothesized mode location relative to the two monotone regression legs: (a) greater than or equal to the right leg and strictly greater than the left leg; (b) in between or equal to either leg, left leg is highest; (c) greater than or equal to the left leg and strictly less than the right leg, right leg is highest; (d) strictly below both, left leg is highest; (e) strictly below both, right leg is highest.

configuration, since the latter is optimal for a less constrained problem and it requires further averaging to produce the optimal fixed mode unimodal regression for the given mode location. Therefore the given mode location cannot be optimal, since we can declare the left black point to be the mode location and this is unimodal at exactly the same configuration (therefore same cost). If the input value at the given hypothesized mode location is actually *equal* to the left black point, then the given hypothesized mode location cannot be a *leftmost* mode location. The same argument holds for Figure 5(c), but for the right black point. Next consider Figure 5(d). Clearly this cannot be an optimal *leftmost* mode location, since any further *ulsrfix* steps will have to average to the left, creating a plateau for which the given mode location is certainly not the *leftmost* point. The conclusion follows by uniqueness of solution to the fixed mode location unimodal regression problem for the given mode location and by correctness of *ulsrfix*.

Finally consider Figure 5(e). For this hypothesized mode location to become a true leftmost mode location, one may average to the right only and the hypothesized mode location level should be raised by at least $\delta + c$, where $c$ is a non-negative constant. This leads to a 'give-in' in terms of fit of at least $(\delta + c)^2$ with respect to the current configuration. However, if one raises the hypothesized mode location level by just $\delta$, one obtains a unimodal configuration of excess give-in of exactly $\delta^2$. Thus the latter configuration is better and the hypothesized leftmost mode location can be safely excluded from consideration.

This leaves the case of Figure 5(a) as the only surviving candidate optimal leftmost mode location and the proof is complete. ■

 

ACKNOWLEDGEMENTS

Our sincere thanks are extended to Henk A. L. Kiers for providing initial software for monotone regression and for helpful comments on the paper, to Sijmen de Jong, Joe B. Kruskal and an anonymous referee for giving insightful comments and suggestions on an earlier version of the manuscript, and to Lars Nørgaard and Carsten Ridder for providing the FIA data. The first author is indebted to Professor Lars Munck, Food Technology, Department of Dairy and Food Science, Royal Veterinary and Agricultural University, Denmark, for financial support through the Nordic Industry Foundation project P93149 and the FØTEK foundation. The second author acknowledges support from the National Science Foundation, through grant NSF EEC 9402384 to the Institute for Systems Research, and the Lockheed-Martin Chair in Systems Engineering, through Professor John Baras. This work would not have been possible without the proliferation of the World-Wide Web. The authors met each other in October 1996 while the first author was surfing the web; this collaboration was entirely over the network and involved the exchange of several hundreds of pieces of e-mail.

## REFERENCES

1. R. A. Harshman, *UCLA Working Papers Phonet.* **16**, 1 (1970).
2. R. A. Harshman, *UCLA Working Papers Phonet.* **22**, 111 (1972).
3. J. B. Kruskal, in *Multiway Data Analyses*, ed. by R. Coppi and S. Bolasco, p. 7, Elsevier/North-Holland, Amsterdam (1989).
4. S. Leurgans, R. T. Ross and R. B. Abel, *SIAM J. Matrix Anal. Appl.* **14**, 1064 (1993).
5. R. Bro, *Chemometrics Intell. Lab. Syst.* **38**, 149 (1997).
6. W. H. Lawton and E. A. Sylvestre, *Technometrics*, **13**, 617 (1971).
7. E. J. Karjalainen and U. P. Karjalainen, *Anal. Chim. Acta*, **250**, 169 (1991).
8. P. J. Gemperline, *Anal. Chem.* **58**, 2656 (1986).
9. F. J. Knorr, H. R. Thorsheim and J. M. Harris, *Anal. Chem.* **53**, 821 (1981).
10. S. D. Frans, M. L. McConnel and J. M. Harris, *Anal. Chem.* **57**, 1552 (1985).
11. Y. Liang and O. M. Kvalheim, *Chemometrics Intell. Lab. Syst.* **20**, 115 (1993).
12. W. Windig, *Chemometrics Intell. Lab. Syst.* **23**, 71 (1994).
13. C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, CAM Vol. 15, SIAM, Philadelphia, PA (1995).
14. R. Bro and S. de Jong, *J. Chemometrics*, **11**, 393 (1997).
15. H. Martens and T. Næs, *Multivariate Calibration*, Wiley, Chichester (1989).
16. P. M. Kroonenberg, *Three-Mode Principal Component Analysis*, DSWO Press, Leiden (1983).
17. B. C. Mitchell and D. S. Burdick, *J. Chemometrics*, **8**, 155 (1994).
18. G. P. H. Styan, *Linear Algebra Appl.* **6**, 217 (1973).
19. R. A. Harshman and M. E. Lundy, *Comput. Statist. Data Anal.* **18**, 39 (1994).
20. W. J. Heiser and P. M. Kroonenberg, *Leiden Psychological Reports, PRM 97–01* (1997).
21. R. E. Barlow, D. J. Bartholomew, J. M. Bremner and H. D. Brunk, *Statistical Inference under Order Restrictions*, Wiley, New York (1972).
22. J. B. Kruskal, *Psychometrika*, **29**, 115 (1964).
23. J. de Leeuw, *Psychometrika*, **42**, 141 (1977).
24. Z. Geng and N. Shi, *Appl. Statist.* **39**, 397 (1990).
25. M. Frisén, *Statistician*, **35**, 479 (1986).
26. N. D. Sidiropoulos and R. Bro, *IEEE Trans. Signal Process.* in press.
27. H. A. L. Kiers, *Psychometrika*, **62**, 251 (1997).
28. D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA (1973).
29. S.-C. Fang and S. Puthenpura, *Linear Optimization and Extensions: Theory and Algorithms*, AT&T/Prentice-Hall, Englewood Cliffs, NJ (1993).
30. R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ (1957).
31. R. Bellman and S. Dreyfus, *Applied Dynamic Programming*, Princeton University Press, Princeton, NJ (1962).
32. S. Dreyfus and A. Law, *The Art and Theory of Dynamic Programming*, Academic, New York (1977).

 

33. D. Bertsekas, *Dynamic Programming and Optimal Control*, Vols I and II, Athena Scientific, Belmont, MA (1995).
34. N. D. Sidiropoulos, *IEEE Trans. Signal Process.* **45**, 389 (1997).
35. N. D. Sidiropoulos, *IEEE Trans. Signal Process.* **44**, 586 (1996).
36. G. W. Ewing, *Instrumental Methods of Chemical Analysis*, McGraw-Hill, New York (1985).
37. R. A. Harshman and W. S. de Sarbo, in *Research Methods for Multimode Data Analysis*, ed. by H. G. Law, C. W. Snyder Jr., J. A. Hattie and R. P. McDonald, Praeger, New York (1984).
38. J. W. Verhoeven, *Pure Appl. Chem.* **68**, 2223 (1996).
39. L. Nørgaard and C. Ridder, *Chemometrics, Intell. Lab. Syst.* **23**, 107 (1994).
40. R. A. Harshman and M. E. Lundy, *Psychometrika*, **61**, 133 (1996).

 