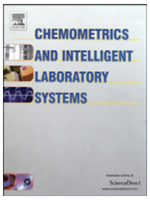
Chemometrics and Intelligent Laboratory Systems 118 (2012) 74–78

Contents lists available at SciVerse ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemolab



Technical details of the equation oriented system

Feng Gan [a,*], Philip K. Hopke [b]

[a] School of Chemistry and Chemical Engineering, Sun Yat-Sen University, Guangzhou 510275, PR China
[b] Department of Chemical and Biomolecular Engineering, Clarkson University, Potsdam, NY 13699, USA

ARTICLE INFO

Article history:
Received 24 April 2012
Received in revised form 2 August 2012
Accepted 2 August 2012
Available online 11 August 2012

Keywords:
Equation oriented system
Matrix decomposition
Conjugate gradient

ABSTRACT

The technical details of equation oriented system are elucidated based on a bilinear model. The key parameters are calculated by applying a revised conjugate gradient to an element of a matrix. It then introduced the major concepts of associated arrays and associated variables in a deductive mean. It shows that the concepts are really useful in developing high efficient algorithm.

1. Introduction

In many chemical studies, the measured properties of the system can be considered to be the linear sum of the term representing the fundamental effects in that system times appropriate weighing factors. For example, the absorbance at a particular wavelength of a mixture of compounds for a fixed path length, z, is considered to be a sum of the absorbencies of the individual components. Multivariate curve resolution (MCR) can extract pure-component spectra from the overlapped spectra of mixtures [1,2]. MCR can be applied to ordinary spectroscopic data [3,4] or data from multiway arrays [5]. Alternatively, MCR was first applied to airborne particle composition data by Henry and Kim [6]. Subsequently they extended their analysis to more than two components [7,8]. MCR has been applied to other environmental data [9]. Thus, MCR has become a widely applied technique for solving the spectrochemical mixture problem.

Many MCR methods have been developed based on the characteristics of the chemical system under study. For example, Gemperline [10] developed iterative target transformation factor analysis (ITTFA) based the assumption of unimodality of peaks in any chromatogram. Based on the evolving characteristics of the components in chemical system or chromatogram, Maeder [11] developed evolving factor analysis, Kvalheim et al. [12,13] developed heuristic evolving latent projections (HELP), and Malinowski [14] developed window factor analysis (WFA). Automated MCR methods have also been developed based on the evolving characteristics [15,16].

Subsequently, new approaches on MCR based on explicit least square analyses have now come into common use. MCR has become most commonly applied using alternating least squares (ALS) algorithms [e.g., 17–21]. Easy to use software has been developed to implement this algorithm [22]. Paatero [23] developed the multilinear engine (ME) as a more general solver that can solve problems that can be expressed as a sum of products. However, ME was relatively slow when resolving large data sets. Thus, Wang and Hopke [24] developed the equation oriented system (EOS) to be able to efficiently resolve large data sets. Both of these approaches are based on a preconditioned conjugate gradient (CG) algorithm [25] and the calculation of the Jacobian matrix. The size of Jacobian matrix can represent computational problems. The ME avoids the problem by constructing an index table to map matrices to vectors. The EOS avoids the calculation of Jacobian matrix by introducing the concept of associated variable, which connects the multidimensional data array with its corresponding vector form in the algorithm. Wang and Hopke hoped to bridge the understanding of vector representation of the original data matrix and the implementation of the array format using the concept. However, the original introduction of the EOS concepts was not sufficiently clear, and thus, the EOS may not have been easy to understand and used.

The aim of this paper is to present the technical details of the EOS. We present the idea of decomposing matrix using a preconditioned CG algorithm. The focus is on the calculation of the key parameters. We introduce the concept of associated variable by applying the CG algorithm to the minimum unit, an element of matrix.

2. Theoretical backgrounds

2.1. Bilinear model

In chemistry, the bilinear model is often used to describe the relation between a measurement and some property quantities. For

[*] Corresponding author.
E-mail address: gan.feng.china@gmail.com (F. Gan).

0169-7439/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.chemolab.2012.08.003

example, in a high-performance liquid chromatography with diode-array detection (HPLC-DAD), following equation is used to describe the measurements of a chemical system of a components:

$$\mathbf{X}_{m\times n} = \mathbf{C}_{m\times a}\mathbf{S}_{a\times n} \tag{1}$$

where $\mathbf{X}_{m\times n}$ is absorbance matrix whose elements are intensities measured at m retention times and n wavelengths; $\mathbf{C}_{m\times a}$ is concentration matrix whose each column is a chromatogram of a single chemical species; $\mathbf{S}_{a\times n}$ is spectral matrix whose each row is a spectrum of the species. Both $\mathbf{C}_{m\times a}$ and $\mathbf{S}_{a\times n}$ are usually unknown for a real chemical system.

Many methods have been developed to directly obtain the solutions for $\mathbf{C}_{m\times a}$ and $\mathbf{S}_{a\times n}$ from the measured $\mathbf{X}_{m\times n}$. These methods usually work well when there are no embedded peaks in the hyphenated chromatography system [26]. In general case, ME and EOS are effective in solving Eq. (1) because they are based on a preconditioned CG algorithm, which iteratively reaches an optimal solution.

2.2. Modified CG algorithm

The original CG algorithm was put forward by Hestenes and Stiefel [27] to solve a system of linear equations $\mathbf{Ax}=\mathbf{b}$, where vector $\mathbf{x}$ is unknown variable and both $\mathbf{A}$ and $\mathbf{b}$ are known variables. The details of the algorithm can be found in a general textbook [25] as a standard method. Paatero [23] preconditioned the algorithm and applied it to matrix decomposition by solving following normal equation:

$$\mathbf{J}^T\mathbf{WJt} = \mathbf{J}^T\mathbf{W}(\mathbf{x}-\mathbf{y}) \tag{2}$$

where $\mathbf{t}$ is step direction vector and the solution of Eq. (2); $\mathbf{J}$ is a Jaccobian matrix which is composed of the estimations of $\mathbf{C}$ and $\mathbf{S}$; $\mathbf{W}$ is a weight matrix; $\mathbf{x}$ is the vector form of $\mathbf{X}$; $\mathbf{y}$ is the vector form of the product of initial $\mathbf{C}$ and $\mathbf{S}$; superscript T denotes matrix transpose.

The details of the modified CG algorithm can be found in the references provided by Paatero [23]. The key of the algorithm is to find the step direction vector $\mathbf{t}$ and step length α, and then update initial $\mathbf{C}$ and $\mathbf{S}$ as follows:

$$\mathbf{f} = \mathbf{f} + \alpha\mathbf{t} \tag{3}$$

where $\mathbf{f}$ is the vector form of the product of $\mathbf{C}$ and $\mathbf{S}$ in the iterative process.

To calculate the $\mathbf{t}$, one calculates the gradient vector $\mathbf{g}=\mathbf{J}^\mathrm{T}\mathbf{W}(\mathbf{x}-\mathbf{y})$, the preconditioning coefficients $z_n=\delta_n p_n g_n$, the step length $\beta=\mathbf{g}^\mathrm{T}\mathbf{z}/\rho$ ($\beta=0$, $\rho=\mathbf{g}^\mathrm{T}\mathbf{z}$), and then update the accumulated direction vector $\mathbf{t}=\beta\mathbf{t}+\mathbf{z}$. The δ_n is the coefficients c_n in step 1.1 of reference [23]. We replace c_n with δ_n because we use character c for the elements of concentration matrix. To calculate the α, one needs to calculate two variables $\tau=\mathbf{t}^\mathrm{T}\mathbf{J}^\mathrm{T}\mathbf{WJt}$ and $\omega=\mathbf{t}^\mathrm{T}\mathbf{J}^\mathrm{T}\mathbf{W}(\mathbf{x}-\mathbf{y})$, and then $\alpha=\omega/\tau$.

Among the calculations, the three parameters, $\rho=\mathbf{g}^\mathrm{T}\mathbf{z}$, $\tau=\mathbf{t}^\mathrm{T}\mathbf{J}^\mathrm{T}\mathbf{WJt}$ and $\omega=\mathbf{t}^\mathrm{T}\mathbf{J}^\mathrm{T}\mathbf{W}(\mathbf{x}-\mathbf{y})$, are primarily concerned because they contain Jacobian matrix $\mathbf{J}$. The ME and EOS took different strategy to avoid the direct calculation of $\mathbf{J}$. In ME, an index table was constructed to search the elements in $\mathbf{J}$. While in EOS, the associated variables were developed to avoid the direct calculation of $\mathbf{J}$. The technical details are revealed by applying the preconditioned CG algorithm to the minimum unit, an element of matrix.

3. Key concepts of EOS

3.1. Associated array and parameter calculations

The element at ith row and jth column of the matrix $\mathbf{X}$ in Eq. (1) is as follows:

$$x_{ij} = C_{i\bullet}S_{\bullet j} \tag{4}$$

where $C_{i.}$ means the ith row of $\mathbf{C}$; $S_{.j}$ means the jth column of $\mathbf{S}$. As the element x_{ij} is a scalar, a simplest model is obtained as follows:

$$x = \mathbf{c}^T\mathbf{s} \tag{5}$$

The subscripts are omitted for simplicity but $\mathbf{c}^\mathrm{T}$ and $\mathbf{s}$ represent the ith row of $\mathbf{C}$ and jth column of $\mathbf{S}$, respectively. Both of them are to be estimated using the modified CG algorithm.

The normal equation was developed in a manner analogous to that used in positive matrix factorization [28]. When small changes are made in both $\mathbf{c}$ and $\mathbf{s}$, say $\Delta\mathbf{c}$ and $\Delta\mathbf{s}$, we get:

$$x = \left(\mathbf{c}^T + \Delta\mathbf{c}^T\right)(\mathbf{s}+\Delta\mathbf{s}) = \mathbf{c}^T\mathbf{s} + \Delta\mathbf{c}^T\mathbf{s} + \Delta\mathbf{c}^T\Delta\mathbf{s} + \mathbf{c}^T\Delta\mathbf{s} \tag{6}$$

If $\Delta\mathbf{c}$ and $\Delta\mathbf{s}$ are small enough, the second-order term can be omitted. Eq. (6) is then rearranged to:

$$\Delta x = \Delta\mathbf{c}^T\mathbf{s} + \mathbf{c}^T\Delta\mathbf{s} = \begin{bmatrix}\mathbf{c}^T & \mathbf{s}^T\end{bmatrix}\begin{bmatrix}\Delta\mathbf{s}\\ \Delta\mathbf{c}\end{bmatrix} \tag{7}$$

where $\Delta x=x-\mathbf{c}^\mathrm{T}\mathbf{s}$. Defining the Jacobian as $\mathbf{J}=[\mathbf{c}^\mathrm{T}\ \mathbf{s}^\mathrm{T}]$, one obtains the following normal equation:

$$\mathbf{J}^T\Delta x = \mathbf{J}^T\mathbf{J}\Delta\mathbf{t}, \qquad \Delta\mathbf{t} = \begin{bmatrix}\Delta\mathbf{s}\\ \Delta\mathbf{c}\end{bmatrix} = \begin{bmatrix}\Delta\mathbf{t}_s\\ \Delta\mathbf{t}_c\end{bmatrix} \tag{8}$$

Here the weight matrix $\mathbf{W}$ is omitted.

The vector $\Delta\mathbf{t}$ is changed in iterative process and it will no longer be directly composed of original $\Delta\mathbf{c}$ and $\Delta\mathbf{s}$. Thus, $\Delta\mathbf{t}$ is presented into two parts, $\Delta\mathbf{t}_\mathrm{s}$ and $\Delta\mathbf{t}_\mathrm{c}$. The $\Delta\mathbf{t}_\mathrm{s}$ is corresponding to $\Delta\mathbf{s}$ and both of them have the same size. In the framework of the EOS, the $\Delta\mathbf{t}_\mathrm{s}$ is the associated array for $\Delta\mathbf{s}$. Similarly, the $\Delta\mathbf{t}_\mathrm{c}$ is the associated array for $\Delta\mathbf{c}$.

In the simplest model, the calculation of $\mathbf{J}\Delta\mathbf{t}$ is quite straightforward as follows:

$$\mathbf{J}\Delta\mathbf{t} = \begin{bmatrix}\mathbf{c}^T & \mathbf{s}^T\end{bmatrix}\begin{bmatrix}\Delta\mathbf{t}_s\\ \Delta\mathbf{t}_c\end{bmatrix} = \mathbf{c}^T\Delta\mathbf{t}_s + \mathbf{s}^T\Delta\mathbf{t}_c \tag{9}$$

Thus, the calculation of τ and ω is quite straightforward too. They are:

$$\tau = (\mathbf{J}\Delta\mathbf{t})^T(\mathbf{J}\Delta\mathbf{t}) = \left(\mathbf{c}^T\Delta\mathbf{t}_s + \mathbf{s}^T\Delta\mathbf{t}_c\right)^2 \tag{10}$$

$$\omega = (\mathbf{J}\Delta\mathbf{t})^T\Delta x = \left(\mathbf{c}^T\Delta\mathbf{t}_s + \mathbf{s}^T\Delta\mathbf{t}_c\right)\Delta x \tag{11}$$

The calculation of ρ requires several steps. First, matrix $\mathbf{J}^\mathrm{T}\mathbf{J}$ should be calculated to obtain its diagonal elements. A simple calculation gives:

$$\mathbf{J}^T\mathbf{J} = \begin{bmatrix}\mathbf{c}\\ \mathbf{s}\end{bmatrix}\begin{bmatrix}\mathbf{c}^T & \mathbf{s}^T\end{bmatrix} = \begin{bmatrix}\mathbf{cc}^T & \mathbf{cs}^T\\ \mathbf{sc}^T & \mathbf{ss}^T\end{bmatrix} \tag{12}$$

So, the vector form of p_k is as follows:

$$\mathbf{p} = \begin{bmatrix}\frac{1}{c_1^2} & \cdots & \frac{1}{c_a^2} & \frac{1}{s_1^2} & \cdots & \frac{1}{s_a^2}\end{bmatrix}^T \tag{13}$$

Second, in the calculation of z_k, δ_c is $\mathbf{c}$ associated array and δ_s is $\mathbf{s}$ associated array. As $\mathbf{g}=\mathbf{J}^\mathrm{T}(\mathbf{x}-\mathbf{y})$ is as follows:

$$\mathbf{g} = \mathbf{J}^T(\mathbf{x}-\mathbf{y}) = \begin{bmatrix}\mathbf{c}\\ \mathbf{s}\end{bmatrix}\Delta x \tag{14}$$

Thus, the vector form of $\mathbf{z}$ is:

$$\mathbf{z} = \begin{bmatrix}\frac{(\delta_c)_1}{c_1} & \cdots & \frac{(\delta_c)_a}{c_a} & \frac{(\delta_s)_1}{s_1} & \cdots & \frac{(\delta_s)_a}{s_a}\end{bmatrix}^T\Delta x \tag{15}$$