Independent Component Analysis Yields Chemically Interpretable Latent Variables in Multivariate Regression

Mats G. Gustafsson*

Uppsala University, Department of Engineering Sciences, Box 528, 751 20 Uppsala, Sweden, and
Department of Genetics and Pathology, Rudbeck Laboratory, 781 85 Uppsala, Sweden

Received April 25, 2005

This work shows that independent component analysis (ICA) can be used to obtain statistically independent and, therefore, chemically interpretable latent variables (LVs) in multivariate regression. Two novel algorithms based on ICA are introduced and compared with two classical methods on simulated data: principal component regression and partial least-squares regression. All methods compared yield accurate predictions, but only those based on ICA yield LVs that are chemically interpretable. Practical limitations of ICA-based regression with respect to the underlying assumptions, sample size, and measurement noise are discussed and illustrated by means of simulations.

1. INTRODUCTION

Linear multivariate regression modeling is the standard approach used in chemistry (chemometrics) for problems such as establishing a quantitative relationship between a large number of molecular properties and an associated molecular activity of interest. Usually, the number of examples available for model design is much fewer than the number of variables (properties), and there are often significant levels of experimental noise present. This results in an ill-posed modeling problem with no unique solution or a solution that is very sensitive to the particular design examples used. Although there exist many methods for ill-posed multivariate regression problems, in chemistry, latent variable (LV) based methods in the form of principal component regression[1] (PCR) and partial least-squares regression[2] (PLSR) have a unique, dominating role. The basic assumption in LV modeling is that the large number of variables used in the multivariate model design are strongly correlated and, therefore, partly redundant. Instead of removing a subset of the variables completely (as in variable subset selection), in LV modeling, all variables are kept and a linear multivariate model is designed for the observed input variables (the molecular properties in the example above) to explicitly model the correlation pattern between the variables. Based on the identified correlations, usually less than a handful of uncorrelated LVs are defined that can be used to reconstruct all important aspects of the original input variables. After this first modeling step has been completed, the small set of extracted LVs are used to build a multivariate regression model that explains the molecular activity of interest.

The dominating roles of LV regression modeling, in general, and PCR and PLSR, in particular, may be explained by the fact that almost all other approaches to ill-posed regression have been suggested by researchers outside chemistry. This includes classical methods such as ridge regression,[3] which yield good predictions but limited insights about the underlying chemical problem.[4] Because LV modeling is based on the idea that there is a small subset of LVs that can explain all important aspects of the large number of strongly correlated variables, the LVs extracted may sometimes offer interesting insights about the key players in a chemical relationship of interest. One illustrative example is the prediction of the protein content in a sample on the basis of an absorption spectrum. In this case, all the spectral components in the multivariate model depend on a single LV, the protein concentration, and are, therefore, very strongly correlated. Successful LV modeling would, in this case, result in a LV model that only contains a single LV that has a one-to-one relationship with the protein concentration. The popularity of the PLS algorithm and, therefore, LV-based modeling in general is presently also beginning to penetrate the area of bioinformatics where PLS has been extensively used in modeling and interpretations of biomolecular interactions and also for tumor classification from gene (mRNA) expression levels; see, for example, the works by Freyhult et al.[5] and Nguyen and Rocke.[6]

Although PCR and PLS are successful and dominating methods in chemistry, a confusing aspect of these LV-based regression methods is that they yield different LVs. One natural question posed by Frank and Friedman[7,8] is, therefore, the following: since both PLS and PCR extract LVs, which ones are the “true” (interpretable) ones? In real world problems, generally, there is no linear LV model that exactly models the observed data, and in these cases, interpretations are very difficult. However, a natural minimum requirement for a LV-based method is that, when the underlying model is exactly linear and there are plenty of training examples, the LVs created should equal the true underlying LVs. In this work, we used a simple numerical example inspired by the prediction of molecular activity from absorption spectra to show that neither the PLS algorithm nor PCR will recover an underlying linear LV model in general. Since the simulations performed resulted in an exact linear model between the input (the spectral components) and the underlying concentrations of the substances involved, ideally, PLS

* Corresponding author e-mail: Mats.Gustafsson@signal.uu.se.

10.1021/ci050146n CCC: $30.25
Published on Web 07/28/2005

or PCR, or both, should result in LVs proportional to the underlying concentrations.

The reason for the failure of both PLS and PCR in this context is related to the rotation ambiguity in classical factor analysis.[9,10] Intuitively, the fundamental problem is that there are infinitely many ways to define LVs that are uncorrelated but there is only one definition that results in LVs that are not only uncorrelated but also independent. As discussed in the context of our earlier work[11] on a probabilistic derivation of the PLS algorithm in terms of stochastic variables, the recent research on independent component analysis (ICA) and independent factor analysis[12,13] show that in order to obtain a unique LV model, it is not enough to consider only second-order statistics (correlation, covariances) between the variables. Since classical factor analysis algorithms as well as PCR and PLSR exclusively use second-order statistics, these methods generally yield different models (from an infinite set of possible models) with uncorrelated rather than independent LVs, see the Appendix. This means that, in general, none of these methods are able to recover a true underlying generative model, even if it is exactly linear.

Motivated by the dominating role of LV-based regression methods (PLSR and PCR) in chemistry and bioinformatics and the observation that neither PLSR nor PCR would, in general, recover true LVs even if samples from an exactly linear generative model is observed, in this work, we introduce a new LV-based regression method called independent component regression (ICR). In ICR, the LV model created is designed to have as statistically independent LVs as possible. By contrast, classical methods such as PCR and PLSR are designed to find LVs that are statistically uncorrelated but not necessarily statistically independent. As will be demonstrated below in a simple numerical example, this approach allows competitive predictions and proper recovery (and therefore proper interpretation) of true LVs when the underlying model is linear.

2. THEORY

2.1. Standard Chemical Multivariate Regression. The standard chemical multivariate regression problem is to model a measured chemical property or response **y** in terms of a set of measured chemical quantities x_k collected in the K-dimensional column vector **x** as

$$\mathbf{y} = \sum_{k=1}^{K} w_k x_k = \mathbf{w}^{\mathrm{T}}\mathbf{x} \tag{1}$$

Here, **w** is a coefficient vector and T denotes the transpose operator that turns the column vector **w** into the row vector $\mathbf{w}^{\mathrm{T}}$. This problem is well-known to be ill-posed when there are fewer examples than coefficients or when there are strong correlations between the components x_k.

In LV-based regression methods such as PCR and PLSR, the idea is to model the strong correlations between the components in **x** explicitly in the form of a LV model[9]

$$\mathbf{x} = \sum_{h=1}^{H} t_k p_h = \mathbf{P}\mathbf{t} \tag{2}$$

where the columns p_h of the $K \times H$ matrix **P** are called loading vectors. In both PCR and PLSR, the H loading vectors are selected such that the LVs t_h, the score values, are uncorrelated. Once the matrix **P** has been obtained and the matrix inverse $(\mathbf{P}^{\mathrm{T}}\mathbf{P})^{-1}$ exists, for each sample **x**, the LVs can be determined as

$$\mathbf{t} = (\mathbf{P}^{\mathrm{T}}\mathbf{P})^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{x} \tag{3}$$

With uncorrelated LVs, the $H \times 1$ coefficient vector $\mathbf{w}_t$ in the reduced linear model

$$\mathbf{y} = \mathbf{w}_t^{\mathrm{T}}\mathbf{t} \tag{4}$$

can safely be obtained using the ordinary least-squares (OLS) criterion

$$V_N(\mathbf{w}_t) = \frac{1}{N}\sum_{n=1}^{N}(y_n - \mathbf{w}_t^{\mathrm{T}}\mathbf{t})^2 \tag{5}$$

where N is the number of training example pairs $(\mathbf{x}_n, y_n)$. This finally yields the input−output model

$$\mathbf{y} = \mathbf{w}_t^{\mathrm{T}}(\mathbf{P}^{\mathrm{T}}\mathbf{P})^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{x} \tag{6}$$

and, thus, the $K \times 1$ coefficient vector $\mathbf{w} = \mathbf{P}(\mathbf{P}^{\mathrm{T}}\mathbf{P})^{-1}\mathbf{w}_t$.

A significant advantage of PLSR in comparison with PCR is the extraction of LVs using not only the input variables in **x** but also the response variable y. In PCR, the LVs are extracted on the basis of a principal component analysis (PCA) of the covariance matrix of the input variables. Only the principal components with the largest variances are used as LVs. Thus, it may result in a LV representation of the input that is based on LVs that explain a large proportion of the inputs but have weak correlations with the response variable y. To avoid this limitation of PCR, a modified form called sorted PCR (SPCR) has been shown to work very well in practice.[14] In SPCR, first, PCA is used to extract uncorrelated candidate variables as in ordinary PCR; then, the extracted variables are sorted according to their correlation with the response variable y. Finally, only the extracted variables with the strongest correlations to y are used as LVs.

2.2. Independent Component Regression. The basic idea of ICR is to first obtain a LV model $\mathbf{x} = \mathbf{P}\mathbf{t}$ of the input **x** in which the LVs t_h ($h = 1, 2, ..., H$) are as statistically independent as possible and then use the LVs to build a reduced model $\mathbf{y} = \mathbf{w}_t^{\mathrm{T}}\mathbf{t}$ as in PCR and PLSR. In the two ICR algorithms presented below, the LV modeling step is performed by a combination of dimensionality reduction based on principal component analysis (PCA) and the application of a particular algorithm for ICA called *FastICA*.

In standard ICA, given N observed samples $\mathbf{z}_n = \mathbf{z}(n)$ of a $L \times 1$ stochastic vector **z**, the problem is to find a square matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_L]$ such that the LV model

$$\mathbf{z} = \sum_{l=1}^{L} s_l \mathbf{a}_l = \mathbf{A}\mathbf{s} \tag{7}$$

where the components of **s** are as statistically independent as possible. An important aspect of standard ICA is that the matrix **A** is square; that is, the number of sources (LVs; the dimension of **s**) equals the number of observations (the dimension of **z**). In other words, standard ICA is not designed for dimensionality reduction but rather for separation (recovery) of independent variables. For this reason, the ICA problem is also known as the blind source separation problem since the components of $\mathbf{s}$ may be regarded as unknown independent sources and the components of $\mathbf{z}$ as observed weighed sums of the sources.

In chemical regression problems, the number of LVs H are usually expected to be relatively few in comparison to the number of components L of the observed vector $\mathbf{z}$. In this paper, the vector $\mathbf{z}$ will equal either the plain input, $\mathbf{z} = \mathbf{x}$, or the concatenation of the input and response, $\mathbf{z} = [\mathbf{x}^T y]^T$. Since $L > H$, standard ICA cannot be applied before the $L \times 1$ observation $\mathbf{z}$ has been reduced to a $H \times 1$ vector $\mathbf{z}_{red}$. In the two ICR algorithms introduced here, this reduction is performed on the basis of PCA, which is used to pick out the H-dimensional subspace in which there is significant variance. The simple idea is that, given H LVs and an $L \times H$ matrix $\mathbf{A}$, then the vectors $\mathbf{z} = \mathbf{As}$ must lie in an H-dimensional subspace of the L-dimensional observation space. Using PCA, this subspace and an associated orthogonal basis are easily found. Assuming that all variables have zero means, using an estimate $\hat{C}_{zz}$ of the covariance matrix $\mathbf{C}_{zz} = E\{\mathbf{zz}^T\}$, the reduced and normalized vector can be expressed as

$$\mathbf{z}_{red} = \Lambda^{-1/2}\mathbf{U}_H^T\mathbf{z} \tag{8}$$

where the hth column u_h of the $L \times H$ matrix $\mathbf{U}_H$ is the eigenvector of $\hat{C}_{zz}$ with eigenvalue λ_h. The normalization matrix $\Lambda^{-1/2}$ is a diagonal matrix with diagonal elements $(\Lambda^{-1/2})_{hh} = 1/\sqrt{\lambda_h}$. This transformation results in a reduced vector $\mathbf{z}_{red}$ with uncorrelated unit variance components and is also well-known as whitening or sphering. The above dimensionality reduction is well-known to yield the best reconstruction of the original variable $\mathbf{z}$. In other words, the reconstruction

$$\hat{\mathbf{z}} = \mathbf{U}_H\mathbf{\Lambda}^{1/2}\mathbf{z}_{red} \tag{9}$$

where the matrix $\Lambda^{1/2}$ is diagonal with diagonal elements $(\Lambda^{1/2})_{hh} = \sqrt{\lambda_h}$ is the best one on average (in the least squares sense).[15]

In the two ICR algorithms, the samples of the reduced vector $\mathbf{z}_{red}$ are used to build a LV model using ICA yielding

$$\mathbf{z}_{red} = \mathbf{A}_{red}\mathbf{s} \tag{10}$$

where $\mathbf{A}_{red}$ is an $H \times H$ matrix and the components s_h of $\mathbf{s}$ are as statistically independent as possible. Substituting this result into the expression for $\mathbf{z}$ in eq 9 yields

$$\mathbf{z} \approx \hat{\mathbf{z}} = \mathbf{U}_H\Lambda^{1/2}\mathbf{A}_{red}\mathbf{s} \tag{11}$$

Introducing the $L \times H$ matrix $\mathbf{A} = \mathbf{U}_H\Lambda^{1/2}\mathbf{A}_{red}$ and assuming that the dimensionality reduction does not introduce any reconstruction error, $\hat{\mathbf{x}} = \mathbf{x}$, then the LV model of $\mathbf{z}$ can be expressed compactly as

$$\mathbf{z} = \mathbf{As} \tag{12}$$

The ICA part of both ICR algorithms is performed using the *FastICA* algorithm, which relies on an approximation of the negentropy (see the Appendix for more details). This algorithm was chosen mainly because of its robustness and computational speed and because of the fact that it is available electronically at http://www.cis.hut.fi/projects/ica/fastica/.

2.3. The Two ICR Algorithms. The first ICR algorithm is called SI−ICR (sorted input ICR) and relies on the idea to replace PCA with ICA in SPCR and is presented in detail in the Appendix. In terms of the formulation of ICA above, this means that the observation vector $\mathbf{z}$ equals the input vector $\mathbf{x}$, that is, $\mathbf{z} = \mathbf{x}$. Note that this means that although the final LV model takes the response $\mathbf{y}$ into account (in the sorting process) like in SPCR, the LVs are estimated (created http://www.cis.hut.fi/projects/ica/fastica/) exclusively using the input $\mathbf{x}$.

In the second ICR algorithm, called SIO−ICR (sorted input−output ICR), the main idea is to estimate the LVs using not only the input $\mathbf{x}$ but also the response $\mathbf{y}$ like in PLSR. In terms of the above formulation of ICA, this means that the observation vector $\mathbf{z}$ equals the concatenation of the input vector $\mathbf{x}$ and the response $\mathbf{y}$, $\mathbf{z} = [\mathbf{x}^T y]^T$.

From the probabilistic perspective of PLS presented in our earlier work,[11] the underlying idea behind PLSR is to simultaneously build two LV models

$$\begin{cases}\mathbf{x} = \mathbf{Pt} + e_x \\ \mathbf{y} = \mathbf{Qu} + e_y\end{cases} \tag{13}$$

of order H, one for the input $\mathbf{x}$ and one for the response $\mathbf{y}$. Here, $\mathbf{t}$ and $\mathbf{u}$ are $H \times 1$ stochastic LV vectors and e_x and e_y are residual errors. $\mathbf{P}$ and $\mathbf{Q}$ are $K \times H$ and $M \times H$ matrices, respectively. Here, we consider the general multiresponse case, but for convenience, the ICR algorithm presented below is designed only for a single scalar response. The LVs $\mathbf{t}$ and $\mathbf{u}$ created by the PLS algorithm are designed to consist of uncorrelated components t_i and u_i, respectively. When the expectation operator E is used, this can be expressed as $E\{t_it_j\} = E\{u_iu_j\} = 0$. The LVs t_i of the input are used for prediction of the LVs u_i using a set of scalar "inner relations"

$$u_i = b_it_i + e_i \tag{14}$$

where b_i is the model coefficients and e_i is the residual error, one for each LV. In vector form, the "inner relations" may be compactly written using a single "inner relation" as

$$\mathbf{u} = \mathbf{Bt} \tag{15}$$

where $\mathbf{B}$ is a diagonal matrix with $B_{ii} = b_i$. The coefficients b_i are found by means of the ordinary least-squares criterion. When the inner relations are used to predict the LVs $\mathbf{u}$, the predicted response is computed as

$$\hat{\mathbf{y}}_p = \mathbf{C}\hat{\mathbf{u}}_p = \mathbf{CBt}_p \tag{16}$$

where $\mathbf{C}$ is a projection matrix, which is one among several outputs from the PLS algorithm. The perhaps unintuitive use of the matrix $\mathbf{C}$ in eq 16 instead of $\mathbf{Q}$ is discussed in detail in ref 11. From eq 13, it is clear that the two separate models of the input and output can be combined into a single LV model by means of the inner relations. Rewriting the model of the response as $\mathbf{y} = \mathbf{QBt} + e_y$, we have

$$\begin{cases}\mathbf{x} = \mathbf{Pt} + e_x \\ \mathbf{y} = \mathbf{QBt} + e_y\end{cases} \tag{17}$$

Collecting the input and response vectors in new vector $\mathbf{z} = [\mathbf{x}^{\mathrm{T}}\mathbf{y}^{\mathrm{T}}]^{\mathrm{T}}$ we obtain the LV model

$$\mathbf{z} = \mathbf{A}\mathbf{t} + e_z \tag{18}$$

where $\mathbf{A} = [\mathbf{P}^{\mathrm{T}}(\mathbf{Q}\mathbf{B})^{\mathrm{T}}]^{\mathrm{T}}$ and $e_z = [e_x{}^{\mathrm{T}}e_y{}^{\mathrm{T}}]^{\mathrm{T}}$. This suggests that one could perform an independent component analysis (ICA) of samples from **z** to recover the LVs in **t**. This approach to LV modeling is significantly different from the one proposed in the I−ICR method above, where the LV model is extracted using ICA based exclusively on the input **x**. In this respect, the LV building process here is more similar to the corresponding process in PLS than that in PCR. In the process of creating the LVs, we get an estimate of the matrix **Q**, which may used for prediction of the response **y** in PLSR. However, since this estimate is part of the solution to an ICA problem, it is not optimized for prediction of the response **y**. Therefore, in the SIO−ICR algorithm, the regression model is designed directly on the basis of the LVs with the explicit aim to minimize the sum of squared prediction errors.

3. SIMULATION RESULTS

To illustrate the performance of ICR and to demonstrate the limited power of PCR and PLS when it comes to interpretations of the LVs extracted, numerical experiments based on samples from a simple linear LV model were performed. These experiments were implemented in MATLAB (Mathworks, U. S. A.) on a standard 1.3 GHz processor.[16] Parts of the simulations relied on functions in the PLS MATLAB toolbox (Eigenvector Research, U.S.A.).

3.1. A Simple Generative LV Model. Consider the following idealized LV model, which is related to the analysis of absorption spectra. This model relies on three assumptions. First, we assume that absorption spectra are collected from solutions consisting of mixtures of H individual substances with unknown individual absorption spectra. Second, we assume that the superposition principle holds; that is, the spectrum collected for one individual mixture is a superposition (weighted sum) of the substance spectra (Beer's law). Finally, we assume that the response (biological, chemical, or other) to each mixture is a fixed weighted sum of the concentrations of the individual substances in the mixture.

The above assumptions can be summarized in a linear multivariate model as follows. With all spectra collected at K wavelengths and stored in data (column) vectors, let the K-dimensional vector **x** denote one stored spectrum of a mixture where each component x_{k} is the absorption measured at the kth wavelength. Similarly, let the K-dimensional vector $\mathbf{a}_h$ denote the unknown spectrum of the hth substance in each mixture. Furthermore, for a given mixture, let the H-dimensional vector **s** be a concentration vector with component s_h equal to the concentration of the hth substance, $h = 1, 2, ..., H$. When the substance spectra in a $K \times H$ matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_H]$ is collected, each observed spectrum **x** may be written as the superposition

$$\mathbf{x} = s_1\mathbf{a}_1 + s_2\mathbf{a}_2 + ... + s_H\mathbf{a}_H = \mathbf{A}\mathbf{s} \tag{19}$$

Assuming measurement noise for each wavelength collected in a K-dimensional vector $\mathbf{e}_x$, we finally obtain the generative LV model

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{e}_x \tag{20}$$

Denoting the response to an individual mixture with spectrum **x** by the scalar y, the assumption above that the response may be expressed as a superposition of the concentrations of the individual substances is

$$y = \mathbf{w}^{\mathrm{T}}\mathbf{s} + e_y = \sum_{h=1}^{H} w_h s_h + e_{\mathrm{y}} \tag{21}$$

where e_y is measurement noise and **w** is an H-dimensional coefficient vector.

Independently of how well the linear LV model in eqs 20 and 21 reflect chemical reality, the resulting LV is interesting because it exactly reflects the structure of the LV models obtained using the classical LV-based PCR and PLSR methods. When a set of training examples generated from the model in eqs 20 and 21 is used, both PCR and PLR would, therefore, produce a LV model $\mathbf{x} = \mathbf{P}\mathbf{t}$ where **P** is an estimate of the matrix **A** above and where **t** is an estimate of the LV vector **s**. Both the PCR and PLSR methods would also yield estimates of the true coefficient vector **w** associated with the response variable y. Ideally, both PLSR and PCR would yield estimates that are nearly identical to the underlying generative model when the number of training examples becomes large.

3.2. Generation of Samples. First, the set of loading vectors in the form of six artificially generated spectra shown in Figure 1 were generated and stored as columns a_i, $i = 1-6$, in a matrix **A**. The spectra were created by uniform sampling of the following mathematical functions at K points (frequencies):

$$a_1(f) = \frac{1}{\sigma_1\sqrt{2\pi}}\exp\left(-\frac{f-f_1}{\sigma_1}\right)^2 \tag{22}$$

$$a_2(f) = \frac{1}{\sigma_{2a}\sqrt{2\pi}}\exp\left(-\frac{f-f_{2a}}{\sigma_{2a}}\right)^2 + \frac{1}{\sigma_{2b}\sqrt{2\pi}}\exp\left(-\frac{f-f_{2b}}{\sigma_{2b}}\right)^2 \tag{23}$$

$$a_3(f) = \frac{1}{\sigma_3\sqrt{2\pi}}\exp\left(-\frac{f-f_3}{\sigma_3}\right)^2 \tag{24}$$

$$a_4(f) = \frac{1}{\sigma_4\sqrt{2\pi}}\exp\left(-\frac{f-f_4}{\sigma_4}\right)^2 \tag{25}$$

$$a_5(f) = \rho(f) \qquad \rho \in U[0, 0.1] \tag{26}$$

$$a_6(f) = \frac{1}{\sigma_{6a}\sqrt{2\pi}}\exp\left(-\frac{f-f_{6a}}{\sigma_{6a}}\right)^2 + \frac{1}{\sigma_{6b}\sqrt{2\pi}}\exp\left(-\frac{f-f_{6b}}{\sigma_{6b}}\right)^2 \tag{27}$$

The fifth spectrum a_5 consists of random numbers ρ drawn from one uniform distribution on the interval [0, 1]. These

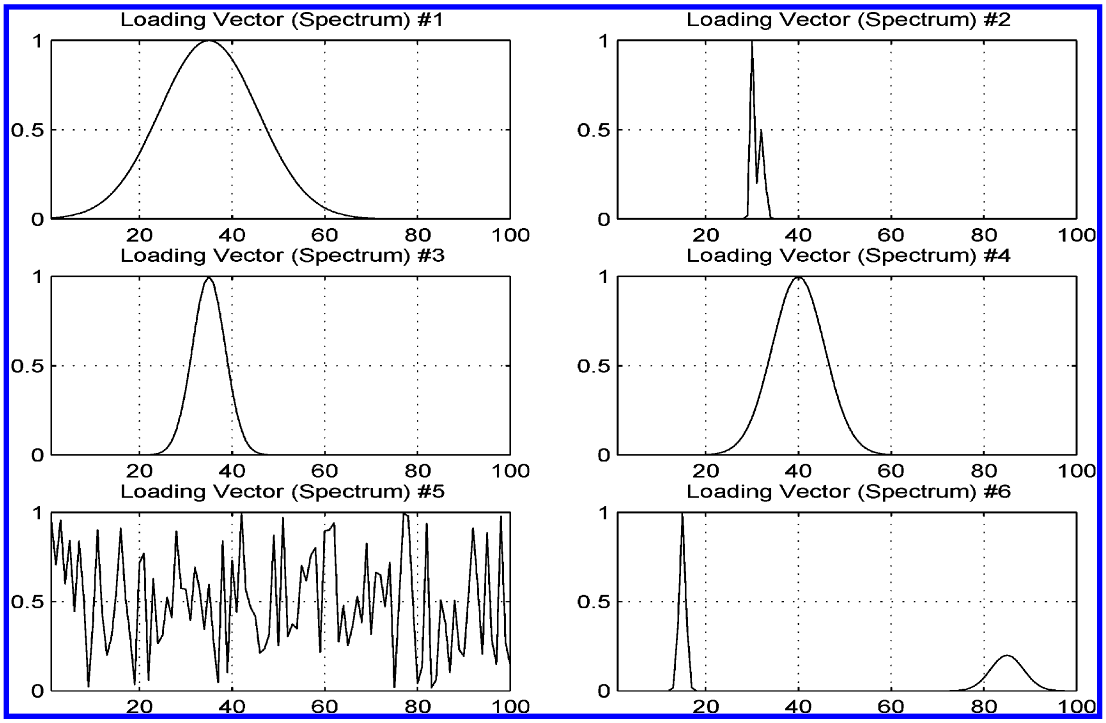

Figure 1. Loading vectors (spectra) of the generative model.

particular spectra were chosen for illustration and convenience and were not intended to reflect any specific chemical experiment. After sampling of these functions, each column a_i was normalized to have unit length.

On the basis of these spectra, examples of spectral mixtures were generated by means of the LV model $\mathbf{x} = \mathbf{As} + \mathbf{e}_x$ in eq 20 where each independent LV s_i was chosen to be uniformly distributed on the interval [0, 1]. The noise vector $\mathbf{e}_x$ was chosen to consist of uncorrelated (independent), normally distributed variables with the same variance $\sigma_x{}^2$.

On the basis of the input samples generated, in this numerical experiment, the responses were then generated as $y = \mathbf{w}^{\mathrm{T}}\mathbf{s} + e_y$ where $\mathbf{w}$ is a coefficient (weight) vector that determines the influence of each LV on the response y and e_y is normally distributed measurement noise.

Remark. Normalization of $\mathbf{a}_i$ is one possibility used here; another would be to allow different intervals for each uniformly distributed LV s_i.

3.3. Parameter Values Used. In the particular experiments presented in this work, the following parameter values were used.

- $K = 100$ (input dimensions).
- $N_{\mathrm{train}} = 500$ spectra (with corresponding responses) for training.
- $N_{\mathrm{test}} = 1000$ spectra (with corresponding responses) for test.
- Spectral parameters: $f_1 = 35$, $\sigma_1 = 15$, $f_{2a} = 30$, $f_{2b} = 32$, $\sigma_{2a} = 0.5$, $\sigma_{2b} = 1 f_3 = 35$, $\sigma_3 = 5$, $f_4 = 40$, $\sigma_4 = 8$, $f_{6a} = 15$, $f_{2b} = 65$, $\sigma_{2a} = 1$, $\sigma_{2b} = 2$.
- Input noise standard deviation $\sigma_x = 0.001$.
- Output noise standard deviation $\sigma_y = 0.001$.
- Response weights: $w_1 = 0$, $w_2 = 0$, $w_3 = 0$, $w_4 = 0.5$, $w_5 = 0.7$, $w_6 = 0.4$.
- Number of test runs used for computation of the average Q^2.

The basic motivations for these parameters are as follows. $N_{\mathrm{train}} = 500$ was chosen to make sure that enough of the training examples were available to obtain interesting illustrative results. As will be discussed later, ICR requires a lot of data. $N_{\mathrm{test}} = 1000$ was chosen to get a good estimate of the predictive performance. The spectral parameters were chosen to obtain nontrivial and partly overlapping spectra. The noise levels $\sigma_x = 0.001$ and $\sigma_y = 0.001$ were included for completeness but set to influence very little. The response weights $w_1 = 0$, $w_2 = 0$, $w_3 = 0$, $w_4 = 0.5$, $w_5 = 0.7$, and $w_6 = 0.4$ were chosen to determine how well the fact that the LVs s_4, s_5, and s_6 alone completely explain the response variable y is reflected in the PLSR, PCR, SPCR, and ICR models: One of the main advantages of PLSR in comparison with PCR pointed out by many has been the ability to extract a simpler model of the input, sifting out only those LVs that explain the response y but not necessarily explain the whole variance of the input.

3.4. Main Experimental Results. *3.4.1. Estimated Loading Vectors.* In Figure 2, the loading vectors of the PLS algorithm (top left) and PCR (top right) obtained from one particular experiment (training set) are presented. Comparing them with the original spectra in Figure 1 shows that the loading vectors (and thus the LVs) are not recovered.

In Figure 2, the loading vectors of the SPCR method (bottom left) and the SI−ICR method (bottom right) are also presented. One should note that the only difference between the loadings in the PCR (top right) and SPCR (bottom left) methods is their relative order, which, in the second case, is corresponding to the correlation coefficient between the corresponding LV and the response variable. In the SI−ICR method (bottom right), we find that very good estimates of the loading vectors associated with the true LVs are yielded. Moreover, the loading vectors are correctly ordered according

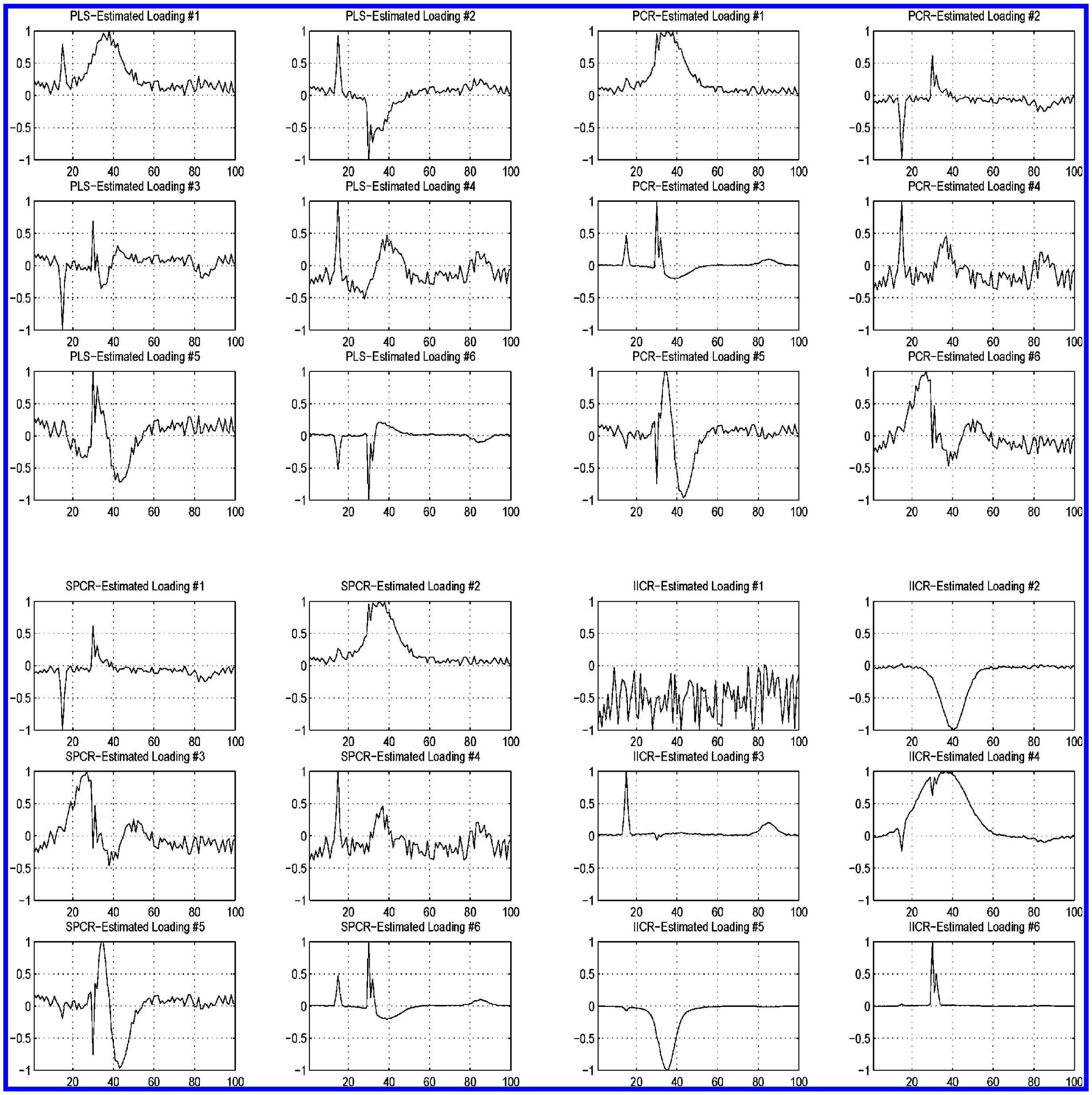

Figure 2. Estimated and normalized loading vectors obtained with PLS (top left), PCR (top right), SPCR (bottom left), and SI−ICR (bottom right). Only the SI−ICR method recovers the true loading vectors in Figure 1.

to the correlation coefficients for the corresponding LV and the response y. From **w**, we know that s_5 is the most influential, then comes s_4 and, finally, s_6. The correct ordering of the LVs in ICR may also be confirmed by the correlation coefficients ρ_h between the LVs and the response variable. In this particular experiment, the magnitudes of the correlation coefficients were $\rho_1 = 0.65$, $\rho_2 = 0.57$, $\rho_3 = 0.49$, $\rho_4 = 0.07$, $\rho_5 = 0.04$, and $\rho_6 = 0.001$, which clearly indicates that the first three estimated loading vectors and their associated LVs are the only ones important for prediction of the response. Essentially the same results as for the SI−ICR method were obtained for the SIO−ICR method and are, therefore, not presented here.

3.4.2. Predictive Performance. The predictive performances of PLSR, PCR, SPCR, SI−ICR, and SIO−ICR are presented in Table 1 on the basis of 100 pairs of training and test sets. The results are presented in terms of the average of the quantity P^2, defined as

$$P^2 = 1 - \frac{\sum_{m\in S_{\mathrm{test}}}[y_{\mathrm{pred}}(m) - y_{\mathrm{obs}}(m)]^2}{\sum_{m\in S_{\mathrm{test}}}[y_{\mathrm{obs}}(m) - y_{\mathrm{mean}}]^2} \qquad (28)$$

S_{test} is the set of indices that belong to the test set, and $y_{\mathrm{obs}}(m)$ is the observed (target) output for test input $\mathbf{x}(m)$. Similarly, $y_{\mathrm{pred}}(m)$ is the predicted output for the test input $\mathbf{x}(m)$. y_{mean} is the average on the training set (here, equal to zero because of mean centering). One should note that the quantity P^2 is based on $N_{\mathrm{test}} = 1000$ independent test examples, whereas the more commonly used quantity Q^2, also known as $R_{\mathrm{CV}}{}^2$, is based on the results from a cross-validation procedure (using the same formula). In the

Table 1. Results with Low Input and Output Measurement Noise Levels ($\sigma_x = 0.001$, $\sigma_y = 0.001$) and Many (500) Training Examples[a]

regression method	P^2 (1 LV)	P^2 (2LVs)	P^2 (3LVs)	P^2 (4LVs)	P^2 (5LVs)
PLSR	0.64	0.84	0.94	0.99	1.00
PCR	0.29	0.68	0.72	0.79	0.82
SPCR	0.37	0.69	0.86	0.94	0.98
SI−ICR	0.52	0.81	0.99	1.00	1.00
SIO−ICR	0.51	0.79	0.99	0.99	1.00

[a] Predictive performance in terms of average P^2 over 100 runs for different numbers of latent variables (LVs).

simulations, the standard deviations of P^2 were typically around 0.04 and were always smaller than 0.09.

The results in Table 1 are based on the average over 100 separate experiments (runs), each with different realizations of the LVs, the measurement noise, and the randomly generated fifth spectrum. The computed average P^2 values confirm earlier findings that SPCR yields a much better performance than conventional PCR for a given number of LVs. Moreover, the results show that PLSR often yield better performance than ICR for one or two LVs, even if ICR extracts the true LVs. This can be explained by the fact that the PLS algorithm builds a LV model designed exclusively for prediction (correlation) with the response variable, not for extraction of the true generative model. This means that the first LVs extracted by PLS might contain linear combinations of true independent LVs that together yield better predictions of the response variable than a smaller subset of these LVs do in ICR.

3.4.3. Summary. In conclusion, we find that the ICR yields an interpretable LV model and predictions comparable with PLSR in this particular case. Moreover, we find that PLSR typically yields better predictions than ICR when the number of LVs are few, and this can be explained by the fact that PLSR is focused more on prediction than correct LV modeling. Finally, we find that the correlation coefficient between a LV and the response variable in an ICR model is useful for selection of the true number of LVs.

Remark. One should note that the P^2 values are computed on the basis of the observed target values, not the true responses. Although the true responses were available in this numerical experiment and, therefore, could have been used when computing P^2, they were ignored as they would not be available in practical real world applications. Anyway, with the very small measurement noise levels used here, adding or ignoring the noise when computing P^2 does not have any significant influence on the results.

4. DISCUSSION

In this paper, two algorithms for ICR are presented that are based on ICA and, in particular, the *FastICA* algorithm.[12] Many alternative ICR algorithms are possible and will be considered in future work. Two problems of particular interest in future work would be to find improvements to the algorithm presented here with respect to the number of training examples needed for good performance and the tolerance against significant levels of measurement noise (also known as the noisy ICA problem). Before discussing these issues, one should note that the linear spectral model used to generate the experimental results using computer simulations may be quite far from chemical reality and that the interpretation in terms of spectral components and concentrations is introduced only to fix ideas and thereby simplify the discussion. More generally, we may assume that each chemical measurement $\mathbf{x}$ is a nonlinear function $\mathbf{g}$ of some real or abstract LVs $\mathbf{s}$, $\mathbf{x} = \mathbf{g}(\mathbf{s})$. Performing a Taylor expansion of the nonlinear function, after truncation of higher order terms and assuming $\mathbf{g}(0) = 0$ (or collecting it in the left-hand side), we obtain the linear approximation

$$\mathbf{g}(\mathbf{s}) = \mathbf{g}(0) + \mathbf{As} + \text{higher order terms} \approx \mathbf{As}$$

where $\mathbf{A}$ is the derivative of $\mathbf{g}$ evaluated at $\mathbf{s} = 0$.

4.1. A Fundamental Theoretical Limitation of ICA. Besides the scaling and permutation ambiguities in ICA discussed in the Appendix, the theory of ICA also shows that in order for the ICA to have a unique (except for scaling and permutation) solution, no more than one of the LVs can be normally distributed. This fact is tightly connected to the well-known rotation ambiguity in classical factor analysis. With more than one Gaussian LV, one can only estimate the ICA model up to a rotation, that is, an orthogonal transformation. This fact is, of course, important in the context of chemical applications and may be summarized as follows: If more than one of the LVs involved in a chemical multivariate problem is Gaussian, not only the classical methods PLSR, PCR, and SPCR will fail to recover the true LVs but this will also be the case for the new ICR method.

4.2. Problems with Small Training Sets. In the numerical examples presented as the main experimental results above, the number of training examples were $N_{\text{train}} = 500$. Although this may become a reasonable number in the future when large scale biological and chemical experiments can be performed, today it is a great number in, for example, a QSAR (quantitative structure−activity relationship) study. Therefore, it is important to note that decreasing the number of training examples significantly reduces the quality of the estimates of the loading vectors in the ICA-based ICR method presented here. This is illustrated for a single run in Figure 3, which shows the loadings obtained with ICR using 150 (left) and 50 (right) training examples.

This is accompanied with the prediction results in Table 2, which, as in Table 1, contains the average of P^2 over 100 runs. The significant decrease in quality of the estimates with a decreasing number of training examples stems from the fundamental fact that estimation of statistical correlations (second-order statistics) is a much simpler problem than estimation of statistical independence, which requires analysis of higher than second-order statistics.

4.3. Problems with Measurement Noise. The existence of estimation errors due to measurement noise is a well-known problem in the context of ICA and is still a very active area of research. Increasing the noise levels in the numerical experiment presented in this paper results in significantly reduced performance for ICR, both in terms of loading vector estimates and predictions. This is illustrated for a single run in Figure 4, which shows the loadings obtained with SIO−ICR using 150 (left) and 50 (right) training examples with the standard deviation of the noise on the input and output

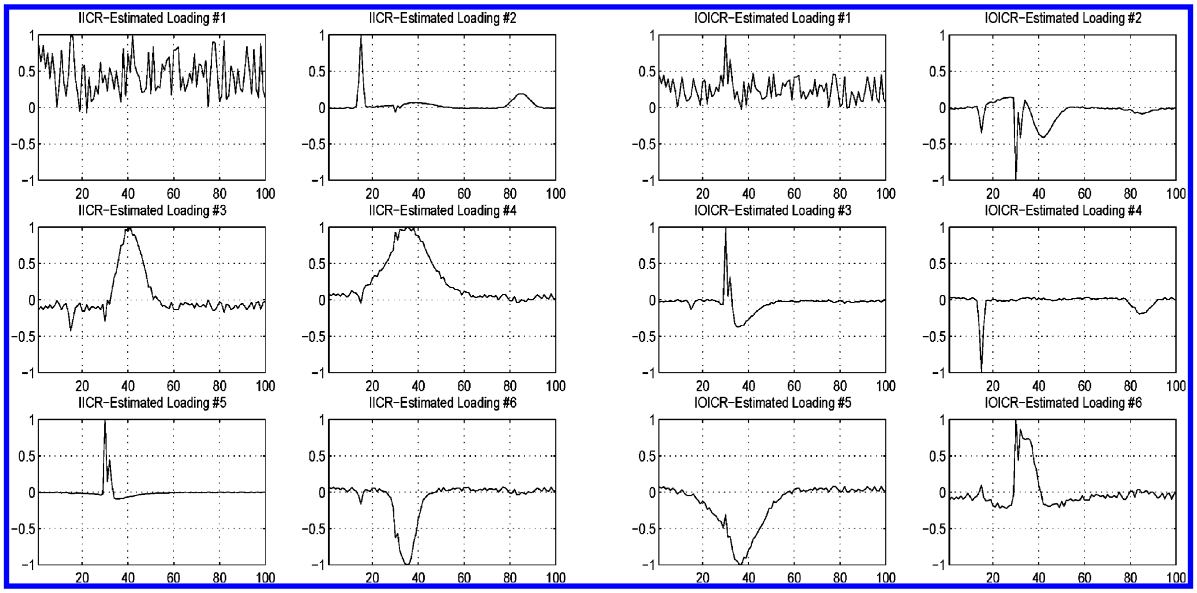

Figure 3. Estimated and normalized loading vectors obtained with SI−ICR (left) and SIO−ICR (right) using low noise levels ($\sigma_x = 0.001$, $\sigma_y = 0.001$) and using 150 (left) and 50 (right) training examples.

Table 2. Results for 150 and 50 Training Examples with Low Input and Output Measurement Noise Levels ($\sigma_x = 0.001$, $\sigma_y = 0.001$)[a]

regression method	P^2 (1 LV)	P^2 (2LVs)	P^2 (3LVs)	P^2 (4LVs)	P^2 (5LVs)
PLSR 150 samples	0.64	0.84	0.94	0.99	1.00
PLSR 50 samples	0.63	0.84	0.93	0.98	1.00
PCR 150 samples	0.29	0.65	0.71	0.79	0.82
PCR 50 samples	0.26	0.58	0.70	0.78	0.82
SPCR 150 samples	0.36	0.66	0.84	0.94	0.98
SPCR 50 samples	0.39	0.65	0.83	0.94	0.98
SI−ICR 150 samples	0.50	0.78	0.96	0.99	1.00
SI−ICR 50 samples	0.47	0.76	0.90	0.97	0.99
SIO−ICR 150 samples	0.48	0.78	0.96	0.98	0.99
SIO−ICR 50 samples	0.43	0.73	0.89	0.96	0.99

[a] Predictive performance in terms of average P^2 over 100 runs for different numbers of latent variables.

Table 3. Results for 150 and 50 Training Examples with the Input and Output Measurement Noise Levels ($\sigma_x = 0.05$ and $\sigma_y = 0.001$)[a]

regression method	P^2 (1 LV)	P^2 (2LVs)	P^2 (3LVs)	P^2 (4LVs)	P^2 (5LVs)
PLSR 150 samples	0.63	0.82	0.90	0.93	0.93
PLSR 50 samples	0.65	0.82	0.88	0.91	0.91
PCR 150 samples	0.28	0.64	0.70	0.77	0.80
PCR 50 samples	0.27	0.59	0.69	0.76	0.79
SPCR 150 samples	0.36	0.65	0.79	0.87	0.91
SPCR 50 samples	0.39	0.63	0.76	0.84	0.88
SI−ICR 150 samples	0.47	0.73	0.88	0.92	0.93
SI−ICR 50 samples	0.44	0.68	0.80	0.86	0.88
SIO−ICR 150 samples	0.47	0.73	0.89	0.92	0.93
SIO−ICR 50 samples	0.44	0.69	0.82	0.90	0.91

[a] Predictive performance in terms of average P^2 over 100 runs for different numbers of latent variables.

increased to $\sigma_x = 0.05$, keeping the output noise level at, as before, $\sigma_y = 0.001$.

This is accompanied with the prediction results in Table 3, which, as in Table 1, contain the average of P^2 over 100 runs. Finally, in Figure 5 and Table 4, the corresponding results using 500 training examples and $\sigma_x = \sigma_y = 0.05$ are presented.

Table 4. Results with the Input and Output Measurement Noise Levels ($\sigma_x = 0.05$, $\sigma_y = 0.05$) and Many (500) Training Examples[a]

regression method	P^2 (1 LV)	P^2 (2LVs)	P^2 (3LVs)	P^2 (4LVs)	P^2 (5LVs)
PLSR	0.61	0.79	0.87	0.90	0.91
PCR	0.28	0.64	0.68	0.74	0.77
SPCR	0.35	0.65	0.78	0.85	0.89
SI−ICR	0.49	0.73	0.89	0.90	0.91
SIO−ICR	0.47	0.72	0.90	0.90	0.91

[a] Predictive performance in terms of average P^2 over 100 runs for different numbers of latent variables (LVs).

5. CONCLUSIONS

Although LV-based regressions in the forms of PLSR and PCR are the dominating tools for ill-posed multivariate regression modeling in chemistry, a simple numerical example presented in this paper demonstrates that the LVs extracted in PLSR and PCR are, in general, not chemically meaningful. This limitation has been explained in terms of the independence of stochastic variables, particularly in the context of a probabilistic interpretation of the PLS algorithm, and the recent theories behind ICA and independent factor analysis.

On the basis of these insights, a new LV-based chemometric regression tool called independent component regression has been introduced. In ICR, the LVs extracted are chosen to be as statistically independent as possible using higher than second-order statistics (correlations, covariances) using a particular algorithm for ICA. In the numerical example presented using a large number of training examples, ICR yielded predictions that were comparable with PLSR and, at the same time, recovered the true LVs of the underlying linear model. When the number of training examples decreased or the measurement noise levels increased, the performance of the particular ICR algorithms presented were shown to decrease significantly, both in terms of prediction and recovery of the underlying LV model.

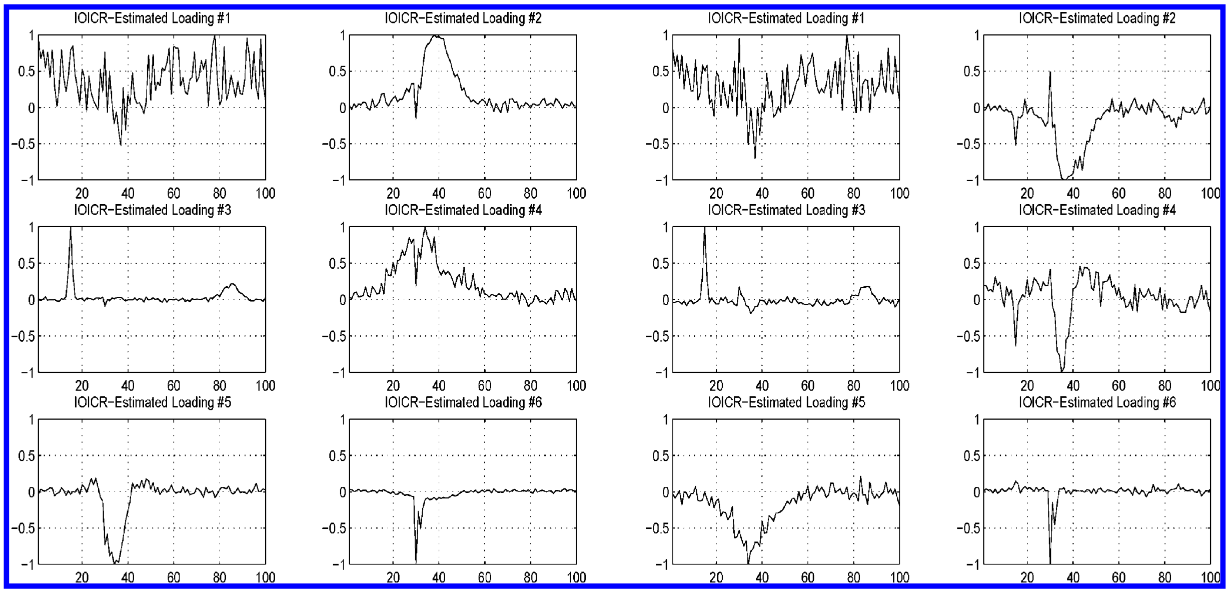

Figure 4. Estimated and normalized loading vectors obtained with SIO−ICR using 150 (left) and 50 (right) training examples. The noise levels were $\sigma_x = 0.05$ and $\sigma_y = 0.001$.

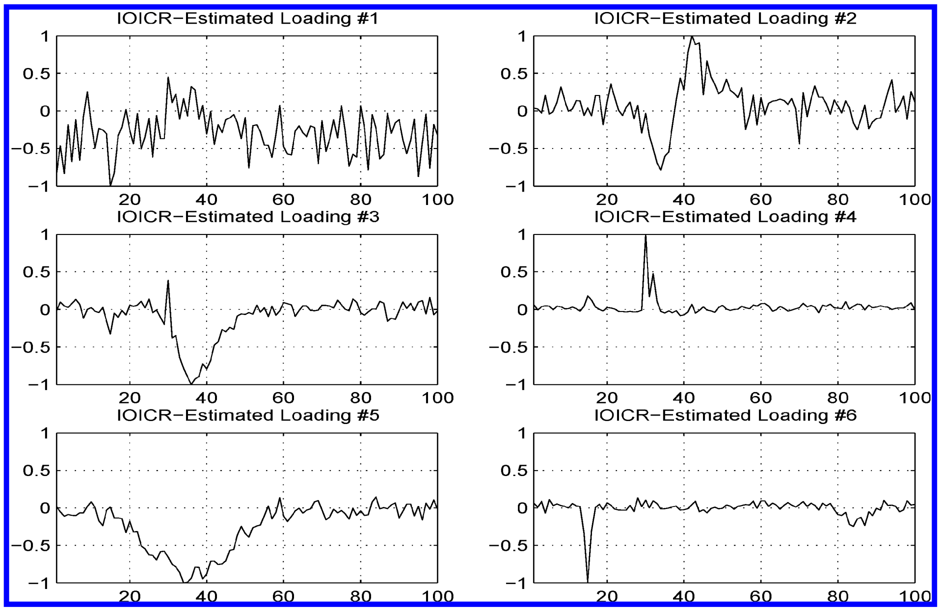

Figure 5. Estimated and normalized loading vectors obtained with SIO−ICR using 500 training examples and the noise levels $\sigma_x = 0.05$ and $\sigma_y = 0.05$.

APPENDIX

A. SI−ICR: An ICR Algorithm Motivated by SPCR. Here, an ICR algorithm very similar to SPCR is introduced. The only significant difference is that the LV model obtained in SPCR using PCA is replaced by a LV model obtained using an algorithm for ICA. The new algorithm is denoted SI−ICR (sorted input ICR) to indicate that the LVs are built exclusively using the input **x** and can be summarized as follows:

Summary of PCR-Motivated ICR Algorithm SI−ICR.

1. Perform a conventional mean centering of all variables.

2. On the basis of the N_{train} mean centered training examples $\mathbf{x}_n$, estimate the covariance matrix using the standard estimate

$$\hat{C}_{xx} = \frac{1}{N_{\text{train}}} \sum_{n=1}^{N_{\text{train}}} \mathbf{x}_n \mathbf{x}_n^{\mathrm{T}}$$

3. Perform a PCA; compute the eigenvectors $\mathbf{u}_k$, $\hat{C}_{xx}\mathbf{u}_k = \lambda_i \mathbf{u}_k$, where λ_k is an eigenvalue, $k = 1, 2, ..., K_{\text{e}}$, and K_{e} is the number of nonzero eigenvalues.

4. Reduce the dimensionality from K to H by transforming the input **x** into the reduced vector $\mathbf{x}_{\text{red}}$ as follows:

$$\mathbf{x}_{\text{red}} = \Lambda^{-1/2} \mathbf{U}_H^{\mathrm{T}} \mathbf{x}$$

$\mathbf{U}_H$ is a $K \times H$ matrix, $\mathbf{U}_H = [u_1, u_2, ..., u_H]$, and $\Lambda^{-1/2}$ is a diagonal $H \times H$ matrix with diagonal elements $(\Lambda^{-1/2})_{hh} = 1/\sqrt{\lambda_h}$, $h = 1, 2, ..., H$. This results in uncorrelated unit variance variables and is well-known as sphering or whitening.

5. Perform an ICA on the reduced vectors using the *FastICA algorithm*, resulting in the following reduced LV (ICA) model:

$$\mathbf{x}_{\text{red}} = \mathbf{A}_{\text{red}}\mathbf{s}$$

where $\mathbf{A}_{\text{red}}$ is the $H \times H$ matrix found by the ICA algorithm and $\mathbf{s}$ is the corresponding estimated LV vector.

6. Estimate the $K \times H$ matrix $\mathbf{A}$ in the nonreduced model in eq 12 as follows: $\hat{\mathbf{A}} = \mathbf{U}\Lambda^{1/2}\mathbf{A}_{\text{red}}$.

7. Identify the loading matrix as $\mathbf{P}_{\text{icr}} = \hat{\mathbf{A}}$ and estimate the LV vector as

$$\hat{\mathbf{s}} = (\mathbf{P}_{\text{icr}}^{\text{T}}\mathbf{P}_{\text{icr}})^{-1}\mathbf{P}_{\text{icr}}^{\text{T}}\mathbf{x}$$

8. For each LV, compute the correlation coefficients ρ_m between $\hat{s}_m$ and the scalar response variable y:

$$\rho_m = \frac{\frac{1}{N_{\text{train}}}\sum_{n=1}^{N_{\text{train}}}\hat{s}_i(n)y(n)}{\hat{\sigma}_i\hat{\sigma}_y}$$

where $\hat{\sigma}_i$ and $\hat{\sigma}_y$ are estimated standard deviations of the LV $\hat{s}_i$ and the response y, respectively.

9. Sort the LVs according to their correlation coefficients ρ_m, yielding a final LV model of the input $\mathbf{x} = \mathbf{P}_{\text{sicr}}\mathbf{t}_{\text{icr}}$ where the columns of $\mathbf{P}_{\text{sicr}}$ are the columns of $\mathbf{P}_{\text{icr}}$ sorted. The components of

$$\mathbf{t}_{\text{sicr}} = (\mathbf{P}_{\text{sicr}}^{\text{T}}\mathbf{P}_{\text{sicr}})^{-1}\mathbf{P}_{\text{sicr}}^{\text{T}}\mathbf{x} = \mathbf{V}_{\text{sicr}}^{\text{T}}\mathbf{x}$$

are the corresponding estimated LVs.

10. For an increasing number H of LVs, create predictive models $y = \mathbf{w}_H^{\text{T}}\mathbf{x}$ of the response y using the ordinary least squares (OLS) criterion, yielding

$$\mathbf{w}_H = \mathbf{V}_H(\mathbf{V}_H^{\text{T}}\mathbf{X}^{\text{T}}\mathbf{X}\mathbf{V}_H)^{-1}\mathbf{V}_H^{\text{T}}\mathbf{X}^{\text{T}}\mathbf{y}_{\text{train}}$$

Here, each row of $\mathbf{X}$ is a realization of the transpose $\mathbf{x}^{\text{T}}$ of the input $\mathbf{x}$ and the H columns $\mathbf{v}_h$ of $\mathbf{V}_H$ are the vectors used to compute the H LVs most correlated with the response variable y. In other words, $\mathbf{V}_H$ consists of the H first columns of $\mathbf{V}_{\text{sicr}}$. The vector $\mathbf{y}_{\text{train}}$ contains the realizations of the response variable in the training set. For more details, see the remark below.

Remark. The determination of the predictive model above can be explained in more detail as follows. For each given number of LVs to be used in the model, compute the LV t_i as $t_i = \mathbf{v}_i^{\text{T}}\mathbf{x}$ where $\mathbf{v}_i^{\text{T}}$ is the ith row in the matrix $\mathbf{V}_{\text{sicr}} = \mathbf{P}_{\text{sica}}(\mathbf{P}_{\text{sica}}^{\text{T}}\mathbf{P}_{\text{sica}})^{-1}$. Collecting the first H vectors $\mathbf{v}_i$ that correspond to the LVs most correlated with the response y as columns in a matrix $\mathbf{V}_H$, we have $\mathbf{t}_H = \mathbf{V}_H^{\text{T}}\mathbf{x}$. Note that all LVs are involved in these calculations. On the basis of the LVs extracted, a predictive OLS model can now be determined as $y = \mathbf{w}_H^{\text{T}}\mathbf{t}_H$, where $\mathbf{w}_H = (\mathbf{T}_H^{\text{T}}T_H)^{-1}\mathbf{T}_H^{\text{T}}\mathbf{y}_{\text{train}}$. Here, each of the N_{train} rows of $\mathbf{T}_H$ is one realization of the column vector $\mathbf{t}_H$ and $\mathbf{y}_{\text{train}}$ is a $N_{\text{train}} \times 1$ vector with the corresponding N_{train} responses. Since $\mathbf{T}_H = \mathbf{X}\mathbf{V}_H$ where the rows of $\mathbf{X}$ are the realizations of the input $\mathbf{x}$, we obtain $\mathbf{w}_H = (\mathbf{V}_H^{\text{T}}\mathbf{X}^{\text{T}}\mathbf{X}\mathbf{V}_H)^{-}\mathbf{V}_H^{1\text{T}}\mathbf{X}^{\text{T}}\mathbf{y}_{\text{train}}$ and $y = \mathbf{t}_H{}^{\text{T}}\mathbf{w}_H = \mathbf{x}^{\text{T}}\mathbf{V}_H(\mathbf{V}_H^{\text{T}}\mathbf{X}^{\text{T}}\mathbf{X}\mathbf{V}_H)^{-1}\mathbf{V}_H^{\text{T}}\mathbf{X}^{\text{T}}\mathbf{y}_{\text{train}}$.

B. SIO−ICR: An ICR Algorithm Motivated by PLSR. The SIO−ICR algorithm can be summarized as follows. For motivations and discussion, see the main text.

1. Perform a conventional mean centering of all variables.

2. Create the new variable $\mathbf{z} = [\mathbf{x}^{\text{T}}\ \mathbf{y}^{\text{T}}]^{\text{T}}$

3. On the basis of the mean centered N_{train} training examples $\mathbf{z}_n$, estimate the covariance matrix using the standard estimate

$$\hat{C}_{\text{zz}} = \frac{1}{N_{\text{train}}}\sum_{n=1}^{N_{\text{train}}}\mathbf{z}_n\mathbf{z}_n{}^{\text{T}}$$

4. Perform a PCA; compute the eigenvectors $\mathbf{u}_k$, $\hat{C}_{zz}\mathbf{u}_k = \lambda_k\mathbf{u}_k$, where λ_k is an eigenvalue, $i = 1, 2, ..., K_{\text{e}}$, and K_{e} is the number of nonzero eigenvalues.

5. Reduce the dimensionality from $K + 1$ to H by transforming $\mathbf{z}$ into the reduced vector $\mathbf{z}_{\text{red}}$ as follows:

$$\mathbf{z}_{\text{red}} = \Lambda^{-1/2}\mathbf{U}_H^{\text{T}}\mathbf{z}$$

$\mathbf{U}_H$ is a $K \times H$ matrix, $\mathbf{U}_H = [u_1, u_2, ..., u_H]$, and $\Lambda^{-1/2}$ is a diagonal $H \times H$ matrix with diagonal elements $(\Lambda^{-1/2})_{hh} = 1/\sqrt{\lambda_h}$, $h = 1, 2, ..., H$. This results in uncorrelated unit variance variables and is well-known as sphering or whitening.

6. Perform an ICA on the reduced vectors using the *FastICA algorithm*, resulting in the following reduced LV (ICA) model:

$$\mathbf{z}_{\text{red}} = \mathbf{A}_{\text{red}}\mathbf{s}$$

where $\mathbf{A}_{\text{red}}$ is the $H \times H$ matrix found by the ICA algorithm and $\mathbf{s}$ is the corresponding estimated LV vector.

7. Estimate the $K + 1 \times H$ matrix $\mathbf{A}$ in the nonreduced model in eqs 20 and 21 as follows: $\hat{\mathbf{A}} = \mathbf{U}\Lambda^{1/2}\mathbf{A}_{\text{red}}$.

8. Identify the $K \times H$ loading matrix as $\mathbf{P}_{\text{icr}}$ by partitioning of the estimate $\hat{\mathbf{A}}$ as $\hat{\mathbf{A}} = [\mathbf{P}_{\text{icr}}^{\text{T}}\mathbf{q}_{\text{icr}}^{\text{T}}]^{\text{T}}$, and estimate the LV vector as

$$\hat{\mathbf{s}} = (\mathbf{P}_{\text{icr}}^{\text{T}}\mathbf{P}_{\text{icr}})^{-1}\mathbf{P}_{\text{icr}}^{\text{T}}\mathbf{x}$$

Note that an indirect estimate of the coefficient vector $\mathbf{w}$ may be obtained from the estimate $\mathbf{q}_{\text{icr}}$ on the basis of the model $y = \mathbf{q}_{\text{icr}}^{\text{T}}\mathbf{s} = \mathbf{q}_{\text{icr}}^{\text{T}}(\mathbf{P}_{\text{icr}}^{\text{T}}\mathbf{P}_{\text{icr}})^{-1}\mathbf{P}_{\text{icr}}^{\text{T}}\mathbf{x}$ but is ignored since the OLS criterion is employed instead (below).

9. For each LV, compute the correlation coefficients ρ_m between $\hat{s}_m$ and the scalar response variable y:

$$\rho_m = \frac{\frac{1}{N_{\text{train}}}\sum_{n=1}^{N_{\text{train}}}\hat{s}_i(n)y(n)}{\hat{\sigma}_i\hat{\sigma}_y}$$

where $\hat{\sigma}_i$ and $\hat{\sigma}_y$ are estimated standard deviations of the LV $\hat{s}_i$ and the response y, respectively.

10. Sort the LVs according to their correlation coefficients ρ_m, yielding a final LV model of the input $\mathbf{x} = \mathbf{P}_{\text{sicr}}\mathbf{t}_{\text{icr}}$ where

the columns of $\mathbf{P}_{\text{sicr}}$ are the columns of $\mathbf{P}_{\text{icr}}$ sorted. The components of

$$\mathbf{t}_{\text{sicr}} = (\mathbf{P}_{\text{sicr}}^{\text{T}}\mathbf{P}_{\text{sicr}})^{-1}\mathbf{P}_{\text{sicr}}^{\text{T}}\mathbf{x} = \mathbf{V}_{\text{sicr}}{}^{\text{T}}\mathbf{x}$$

are the corresponding estimated LVs.

11. For an increasing number H of LVs, create predictive models $y = \mathbf{w}_H^{\text{T}}\mathbf{x}$ of the response y using the OLS criterion, yielding

$$\mathbf{w}_H = \mathbf{V}_H(\mathbf{V}_H^{\text{T}}\mathbf{X}^{\text{T}}\mathbf{X}\mathbf{V}_H)^{-1}\mathbf{V}_H^{\text{T}}\mathbf{X}^{\text{T}}\mathbf{y}_{\text{train}}$$

Here, each row of $\mathbf{X}$ is a realization of the transpose $\mathbf{x}^{\text{T}}$ of the input $\mathbf{x}$ and the H columns $\mathbf{v}_h$ of $\mathbf{V}_H$ are the vectors used to compute the H LVs most correlated with the response variable y. The vector $\mathbf{y}_{\text{train}}$ contains the realizations of the response variable in the training set. For more details about the last two formulas, see the remark after the summary of the SI−ICR algorithm above.

C. Basic Ideas Behind Algorithms for ICA. Consider two zero mean stochastic variables x and y with probability density functions (pdf's) $f_x(x)$ and $f_y(y)$. Also assume that x and y are *not* statistically independent. This means that their joint pdf $f_{xy}(x,y)$ does not equal the product of the individual (marginal) pdf's, $f_{\text{x,y}}(x,y) \neq f_x(x)f_y(y)$. Consider the two stochastic variables $x = \xi$ and $y = \xi^2 - 1$, where ξ is a normally distributed stochastic variable with zero mean and unit variance. Then, one can easily show that, although x and y rely on only one common LV ξ, they are completely uncorrelated:

$$E\{xy\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xy\, f_{\text{x,y}}(x,y)\, \text{d}x\, \text{d}y = 0$$

This example indicates that the transformation of correlated variables (like in PLSR and PCR) to obtain uncorrelated LVs does not, in general, mean that the new uncorrelated LVs are also independent. For Gaussian variables, the equivalence between being uncorrelated and independent is well-known, but this is a special case not valid for other, non-Gaussian, stochastic variables.

The difference between uncorrelated and independent stochastic variables can also be illustrated by the following example. Consider two stochastic variables x and y, which are projections of a zero mean multidimensional stochastic variable $\mathbf{z}$, $x = \mathbf{w}_x^{\text{T}}\mathbf{z}$ and $y = \mathbf{w}_y^{\text{T}}\mathbf{z}$. Then,

$$E\{xy\} = \mathbf{w}_x^{\text{T}}\mathbf{C}_{zz}\mathbf{w}_y$$

where $\mathbf{C}_{zz}$ is the covariance matrix of $\mathbf{z}$. Thus, x and y are uncorrelated for all projection vectors satisfying the equality $\mathbf{w}_x^{\text{T}}\mathbf{C}_{zz}\mathbf{w}_y = 0$. Apparently, there are infinitely many such projections, but there is only one projection that yields independent variables x and y (provided that the variables are non-Gaussian).

Assuming that $\mathbf{z}$ is two-dimensional and has been generated as $\mathbf{z} = \mathbf{A}\mathbf{s}$, where the 2 × 2 matrix $\mathbf{A}$ is invertible and the components of $\mathbf{s}$ are independent, then, there is a subset of projection vectors that makes x and y proportional to the two components s_1 and s_2 in $\mathbf{s}$. All these projection vectors are characterized by the fact that the resulting projections x and y are not only uncorrelated but also statistically independent. The goal of independent component analysis is to recover one pair of these projection vectors.

ICA algorithms are designed to find a square matrix $\mathbf{W}_{\text{ica}}$, which, after multiplication with $\mathbf{x}$, yields a response $\mathbf{z} = \mathbf{W}_{\text{ica}}\mathbf{x}$ where the components of $\mathbf{z}$ should be as independent as possible. If the underlying model is exactly linear as in eq 19, there exists a matrix that makes all components of $\mathbf{z}$ independent, and one can show that the matrix must have the following structure

$$\mathbf{W}_{\text{ica}} = \mathbf{P}_{\text{perm}}\mathbf{D}\mathbf{A}^{-1} \tag{29}$$

where $\mathbf{D}$ is a diagonal scaling matrix and $\mathbf{P}_{\text{perm}}$ is a permutations matrix that permutes the elements of a vector when multiplication is performed. Thus, $\mathbf{z} = \mathbf{P}_{\text{perm}}\mathbf{D}\mathbf{A}^{-1}\mathbf{x} = \mathbf{P}_{\text{perm}}\mathbf{D}\mathbf{A}^{-1}\mathbf{A}\mathbf{s} = \mathbf{P}\mathbf{D}\mathbf{s}$, which shows that the variables in $\mathbf{z}$ now are scaled versions of the original variables in **s,** which also may have been permuted (reordered) by the permutation matrix $\mathbf{P}_{\text{perm}}$. In conclusion, the ICA problem cannot be solved uniquely; there will always be ambiguity with respect to order and scaling. Fortunately, this is not of any importance in the first ICR algorithm presented below, which, after the ICA step, sorts the independent LVs extracted according to their correlation coefficients with the response variable y like in SPCR and then uses them to build a linear model of $\mathbf{y}$.

Several families of algorithms for ICA have been proposed recently that rely on different theoretical concepts.[12] One approach is to choose the rows of $\mathbf{W}_{\text{ica}}$ in such a way that the resulting projections in $\mathbf{z}$ become as non-Gaussian as possible. Here, different measures of non-Gaussianity have been used such as kurtosis and negentropy. Another approach is to choose $\mathbf{W}_{\text{ica}}$ in such a way that the theoretical mutual information[17] between the components of $\mathbf{z}$ becomes as small as possible. A third approach to ICA algorithms has been a maximum likelihood estimation where non-Gaussian probability density functions are assigned to each LV.

ACKNOWLEDGMENT

This work was supported by the Wallenberg Consortium North, The Swedish Research Council (VR), the Göran Gustafsson Foundation, Carl Tryggers stiftelse, and the Faculty of Science and Technology (Uppsala University). Many thanks also to Dr. Jörgen Jonsson (then, at Eurona Medical and Amersham Biosciences, Uppsala) for interesting discussions about partial least-squares regression and independent latent variables.

REFERENCES AND NOTES

(1) Massy, W. Principal component regression in exploratory statistical research. *Am. Stat.* **1965**, *60*, 234−246.

(2) Geladi, P.; Kowalski, B. Partial least-squares regression: A tutorial. *Anal. Chim. Acta* **1986**, *185*, 1−17.

(3) Hoerl, A.; Kennard, R. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* **1970**, *12*, 55−67.

(4) Wold, S. Discussion: PLS in chemical practice. *Technometrics* **1993**, *35*, 136−139.

(5) Freyhult, E.; Prusis, P.; Lapinsh, M.; Wikberg, J.; Moulton, V.; Gustafsson, M. Unbiased descriptor and parameter selection confirms the potential of proteochemometric modelling. *BMC Bioinf.* **2005**, *6* (1), 50.

(6) Nguyen, D.; Rocke, D. Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics* **2002**, *18*, 39−50.

(7) Frank, I.; Friedman, J. Statistical View of Chemometrics Regression Tools. *Technometrics* **1993**, *35*, 109−135.

(8) Frank, I.; Friedman, J. Response. *Technometrics* **1993**, *35*, 143−148.

(9) Everitt, B. *An Introduction to Latent Variable Models*; Chapman and Hall: London, 1984.

(10) Johnson, R.; Wichern, D. *Applied Multivariate Statistical Analysis*; Prentice Hall: Upper Saddle River, NJ, 1998.

(11) Gustafsson, M. A probabilistic derivation of the PLS algorithm. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 288−294.

(12) Hyvärinen, A.; Oja, E. Independent component analysis: Algorithms and applications. *Neural Networks* **2000**, *13*, 411−430.

(13) Attias, H. Independent factor analysis. *Neural Computation* **1998**, *11*, 803−851.

(14) Egan, W.; Brewer, W.; Morgan, S. Measurement of carboxyhemoglobin in forensic blood samples using UV/vis spectrometry and improved principal component regression. *Appl. Spectrosc.* **1999**, *53* (2), 218−225.

(15) Fukunaga, K. *Introduction to Statistical Pattern Recognition*; Academic Press: San Diego, CA, 1990.

(16) The MATLAB code used in the numerical experiments is available from the author upon request. E-mail: mg@signal.uu.se.

(17) Cover, T. *Elements on Information Theory*; John Wiley & Sons: New York, 1991.

CI050146N