Chemometrics and Intelligent Laboratory Systems 119 (2012) 21–31

Contents lists available at SciVerse ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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

A tutorial on the Lasso approach to sparse modeling

Morten Arendt Rasmussen *, Rasmus Bro

Copenhagen University, Rolighedsvej 30, 1958 Frederiksberg C, Denmark

ARTICLE INFO

Article history:
Received 13 December 2011
Received in revised form 3 October 2012
Accepted 5 October 2012
Available online 13 October 2012

Keywords:
Sparsity
L_1 norm
(Bi)convex optimization
Lasso

ABSTRACT

In applied research data are often collected from sources with a high dimensional multivariate output. Analysis of such data is composed of e.g. extraction and characterization of underlying patterns, and often with the aim of finding a small subset of significant variables or features. Variable and feature selection is well-established in the area of regression, whereas for other types of models this seems more difficult. Penalization of the L_1 norm provides an interesting avenue for such a problem, as it produces a sparse solution and hence embeds variable selection. In this paper a brief introduction to the mathematical properties of using the L_1 norm as a penalty is given. Examples of models extended with L_1 norm penalties/constraints are presented. The examples include PCA modeling with sparse loadings which enhance interpretability of single components. Sparse inverse covariance matrix estimation is used to unravel which variables are affecting each other, and a modified PCA to model data with (piecewise) constant responses in e.g. process monitoring is shown. All examples are demonstrated on real or synthetic data. The results indicate that sparse solutions, when appropriate, can enhance model interpretability.

1. Introduction

Data analytical problems are becoming more and more complex, with more challenging questions and with data of increasing complexity for unraveling of these questions. For example, one may be interested in analyzing a data set with information of a large number of metabolites. Ordinary data analysis tools, such as e.g. principal component analysis, will typically lead to a huge number of variables influencing the model significantly. Hence it is difficult to assess which variables are most important for further investigation. One approach to reduce complexity is by forcing less influential variables to have zero influence on the model. This can be achieved by modifying e.g. a regression model such that rather than finding a regression vector that only provides good predictions, a well-predicting regression vector is sought that is also *sparse* (has many zero regression coefficients). Sparse and sparsity are used throughout this work for characterization of the model parameter and not to be misinterpreted as sparse data structures (raw data where values are null).

The concept of using sparsity actively for achieving *simpler* models has received huge attention within fields such as statistical learning, data mining and signal processing [1–4]. This has been done as a means to achieve more interpretable solutions but also because in many applications the *real solution* (i.e. the underlying process that generated data) is truly sparse.

Imagine a linear regression problem $\mathbf{y}=\mathbf{Xb}+\mathbf{e}$, where $\mathbf{X}$ $(n\times p)$ and $\mathbf{y}$ $(n\times 1)$ are known and $\mathbf{b}$ $(p\times 1)$ is unknown. The aim is to find an estimate of the regression vector $(\hat{\mathbf{b}})$ with good predictive performance which is sparse (a number of elements of $\hat{\mathbf{b}}$ are exactly zero). There exists a variety of methods for achieving sparse solutions: setting *small* coefficients of $\hat{\mathbf{b}}$ to zero (hard thresholding), forward stepwise addition of variables that increase performance the most, backward elimination of the least significant parameters, etc. In this paper we focus on the use of a so called L_1 norm penalty as a means to obtain sparse solutions not only in regression problems but in many kinds of multivariate models. The L_1 norm refers to the sum of absolute values of a vector.

In 1996, Tibshirani [5] published the method least absolute shrinkage and selection operator, also known as the Lasso, which is the most important method using an L_1 norm constraint for regression purposes. Let $\mathbf{X}$ $(n\times p)$ and $\mathbf{y}$ $(n\times 1)$ be known and $\mathbf{b}$ $(p\times 1)$ be unknown forming the linear regression problem: $\mathbf{y}=\mathbf{Xb}+\mathbf{e}$. The unbiased ordinary least squares (OLS) solution is the solution to: $\arg\min_b\left(\|\mathbf{y}-\mathbf{Xb}\|_2^2\right)$, which has the solution $\hat{\mathbf{b}}=\mathbf{b}_{OLS}=\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$. Often this model overfits and hence has bad predictive performance when applied to new data. Constraining the solution is one way to deal with this issue and the Lasso is one way to constrain a regression vector estimate. The Lasso can be formulated as the constrained optimization problem in Eq. (1). The scalar c is a tuning parameter just like the number of components in a PLS model. For an appropriate bound (c) this returns a sparse solution [5]. *Pure* variable selection, where only the number of active variables is of interest, can be seen as penalizing the L_0-norm of the regression coefficients also referred to as the cardinality (number of

* Corresponding author. Tel.: +45 35 33 31 97.
E-mail addresses: mortenr@life.ku.dk (M.A. Rasmussen), rb@life.ku.dk (R. Bro).

0169-7439/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.chemolab.2012.10.003

parameters $\neq 0$). This problem is not convex, and is therefore practically impossible to solve for large number of parameters. Penalizing the L_1 norm is the *closest* convex relaxation/alternative [1].

$$\arg\min_{\mathbf{b}} \left(||\mathbf{y}-\mathbf{X}\mathbf{b}||_2^2 \right) \text{subject to} ||\mathbf{b}||_1^1 \leq c. \quad (1)$$

Here $||\cdot||_2^2$ refers to the sum of squared elements of the vector — the (squared) L_2 norm and $||\cdot||_1^1$ is the sum of the absolute values of the vector — the L_1 norm. See Eqs. (2) and (3).

$$||\mathbf{x}||_2^2 = \sum_{i=1,\ldots,n} x_i^2 \quad (2)$$

$$||\mathbf{x}||_1^1 = \sum_{i=1,\ldots,n} |x_i| \quad (3)$$

The first part of the optimization problem (Eq. (1)) is the squared residuals, and if c is chosen 'large enough', such that the constraint ($||\mathbf{b}||_1^1 \leq c$) is not active, the solution is identical to the OLS solution. Selecting smaller values for c, this solution $\left(\hat{\mathbf{b}} = \mathbf{b}_{\text{OLS}}\right)$ is no longer valid as the length of $\hat{\mathbf{b}}$ is *too long* in an L_1 norm sense.

For any given c in the range between zero and the L_1 norm of the OLS solution ($0 \leq c \leq ||\mathbf{b}_{\text{OLS}}||_1^1$), there exists a unique solution. It turns out that this solution is sparse and more so, the smaller c is. This is the crux of the use of the L_1 norm penalty and will be explained in detail in Section 2.

For ill posed problems, the Lasso is an alternative to other methods such as ridge regression, partial least squares (PLS) regression and principal component regression (PCR). Contrary to Lasso, ridge regression, PLS and PCR produce dense solutions, that is; regression vectors with all elements being non-zero.

The framework of bounding the L_1 norm of some entity in order to achieve sparsity is not restricted to regression type problems, and there exist variants of a wide range of models where minimization of a least squares criterion is extended with an L_1 norm constraint on (some of) the parameters of the model. Sparse principal component analysis [6–8], sparse PLS [9], sparse canonical variate analysis [8], sparse linear discriminant analysis [10], fused Lasso [11] and sparse support vector machine [12] are examples of such.

The present work will present methods often used in chemometrics but modified with an L_1 penalty in order to achieve sparsity.

The paper is organized with a general introduction to why solutions from minimization/bounding of the L_1 norm leads to sparse results. Four examples where different methods are used to model real/synthetic data are presented. In Appendix 1, some representative implementations for solving L_1 norm penalized problems are given.

2. L_1 bounding/penalization

In order to achieve some intuitive understanding of why bounding of the L_1 norm returns a sparse solution, a toy example in two dimensions is presented. Imagine a regression problem with two independent variables and their corresponding regression coefficients b_1 and b_2 respectively ($\mathbf{b} = [b_1 b_2]$). Before looking at the L_1 norm we start by exploring the L_2 norm bound. This has a rather similar form as Eq. (1):

$$\arg\min_{b} \left(||\mathbf{y}-\mathbf{X}\mathbf{b}||_2^2 \right) \text{subject to} \mathbf{b}_2^2 \leq c. \quad (4)$$

The feasible solutions for this problem is set by $||\mathbf{b}||_2^2 \leq c$ which is a disk (or a ball for higher dimensions) around the origin with a diameter of $\sqrt{c}$. A geometrical representation of this parameter space is shown in Fig. 1 with the L_2 norm constraint being the blue ball. The OLS solution $\left(\hat{\mathbf{b}}_{\text{OLS}}\right)$, by definition, has the minimum squared error loss, and as the distance to $\hat{\mathbf{b}}_{\text{OLS}}$ increases, the loss increases as well. This is shown as iso-loss contours around $\hat{\mathbf{b}}_{\text{OLS}}$. The optimal solution (to Eq. (4)) is the point where these loss contours touch the feasible set of solutions. Due to the shape of the L_2 ball, a solution with parameters set to exactly zero rarely occurs. Applying the L_2 norm constraint on regression problems, will thus shrink the coefficients towards zero norm with increasing penalty (decreasing c). However, only for $c = 0$ will the coefficients be exactly zero. Penalizing the L_2 norm of the regression vector, is known as Ridge regression [13].

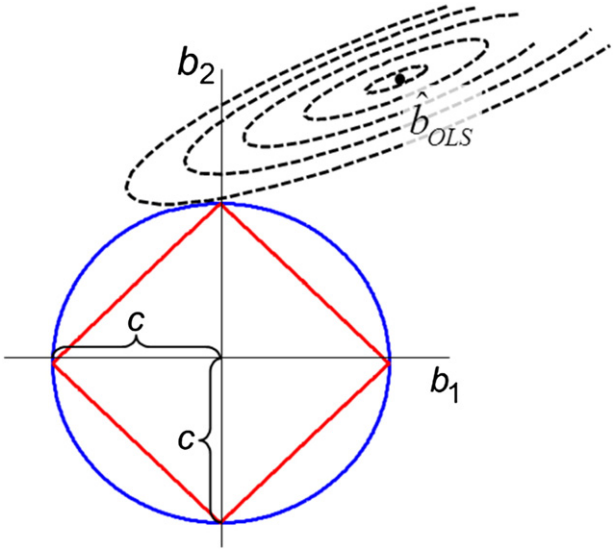

Fig. 1. Two dimensional parameter space (b_1 vs. b_2), $\hat{b}_{\text{OLS}}$ is the (unconstrained) ordinary least squares estimate, the contours reflect the estimates of b with equal deviation in terms of squared error loss. The ridge and Lasso constraints are shown as an L_2 ball (blue circle) and an L_1 ball (red square) respectively.

Turning to L_1 norm constraints, the L_1 *ball* is a (hyper) square/cube with corners on the coordinate axes where all but one parameter is exactly zero (see Fig. 1). It is geometrically easy to see that the loss contours (almost) always touches the hyper cube in a corner or on an edge between corners with some of the parameters being exactly zero [5]. Thus, when the L_1 norm constraint is active, it will lead to some regression coefficients being exactly zero as opposed to the L_2 constraint which will just shrink the norm of the whole regression vector without forcing specific elements to zero.

2.1. Some motivating examples

We wish to solve a regression problem using Lasso (Eq. (1)), and explore the regression coefficients for different magnitudes of the penalty. For the purpose, spectral NIR data ($n = 655$, $p = 15$) measured on pharmaceutical tablets with *hardness* of the tablets as the independent variable are used. Be aware that only 15 variables (out of 650) are used to demonstrate the Lasso (see Section 2.2.1 for the reason why this variable reduction is done). For details on data see the Chambersburg Shoot-out 2002 data set [http://www.idrc-chambersburg.org/shootout_2002.htm — December, 13th 2011].

The Lasso solution corresponds to the OLS solution (for $n > p$) in the cases where the constraint (the bound on the sum of the absolute values of the regression coefficients c in Eq. (1)), is *higher* than the sum of the absolute values of the OLS solution. Tightening of this bound moves the solution away from the OLS solution, and in the case of $c = 0$, all coefficients are zero and there is *no* meaningful solution. There is a path of solutions (known as the *solution path*) for different values of c between these two extremes. In Fig. 2, the regression coefficients for four Lasso solutions and the OLS solution corresponding to parts of the solution path are shown.

In order to select *one* solution, the tuning parameter c is selected in a similar fashion as selecting the number of components in a PLS model,

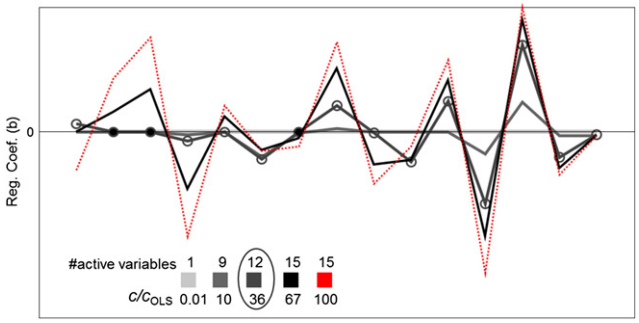

Fig. 2. Sequence of regression coefficients estimated via the Lasso (gray-scale) and OLS solution (red and dashed). The color code is extended with the *number of active variables* above and *relative* L_1 *norm* (c/c_{OLS}) compared to the OLS estimates below. The optimal solution in terms of minimum crossvalidation error has 12 active variables. This solution is highlighted with circles (○ — active, ● — passive).

for example by crossvalidation. Depending on c, some of the variables are deselected (regression coefficients = 0) and are hence deemed *not important* for prediction. This is how Lasso solutions effectively lead to variable selection through sparse regression vectors. For the present example the optimal solution in terms of minimum crossvalidated error is a solution with 12 active and three passive variables (see Fig. 2).

The Lasso is a *least squares loss* criterion associated with regression and extended with an L_1 constraint on the regression vector. This way of extending a loss function with a constraint is not restricted to regression types of problems, and can in principle be formulated for all types of models which have a well-defined loss function. Consider for example the following two models.

PCA

PCA on $\mathbf{X}$ $(n \times p)$ with k components.

$$\underset{\mathbf{T},\mathbf{P}}{\arg\min} \left(\|\mathbf{X} - \mathbf{T}\mathbf{P}^{\mathbf{T}}\|_F^2 \right)$$

$$\text{subject to} \|\mathbf{t}_i\|_1^1 \le c_{t_i}, \|\mathbf{p}_i\|_1^1 \le c_{p_i} \text{ for } i = 1, \ldots, k$$

where $\|\cdot\|_F^2$ is the Frobenius norm of the matrix (sum of squared elements) and $\mathbf{t}_i$ and $\mathbf{p}_i$ are the columns of $\mathbf{T}$ and $\mathbf{P}$ respectively. This gives a solution with sparsity imposed on the scores and the loadings.

PARAFAC

PARAFAC on the tensor $\mathbf{X}$ $(n \times p \times q)$ with k components

$$\underset{\mathbf{A},\mathbf{B},}{\arg\min}\, \mathbf{C} \left(\| \underline{\mathbf{X}} - \mathbf{A}(\mathbf{C} |\otimes| \mathbf{B}) \|_F^2 \right)$$

subject to $\|\mathbf{a}_i\|_1^1 \le c_{a_i}$, $\|\mathbf{b}_i\|_1^1 \le c_{b_i}$, and $\|\mathbf{c}_i\|_1^1 \le c_{c_i}$, for $i = 1, \ldots, k$ where $|\otimes|$ is the Khatri-Rao product of two matrices with the same number of columns, $\mathbf{X}$ $(n \times pq)$ is the unfolded $\mathbf{X}$, and $\mathbf{a}_i$, $\mathbf{b}_i$ and $\mathbf{c}_i$ are the columns of $\mathbf{A}$ $(n \times k)$, $\mathbf{B}$ $(p \times k)$ and $\mathbf{C}$ $(q \times k)$ respectively. This returns a solution with sparsity in all three modes.

Algorithms for numerical estimation of solutions are well established for regression types of problems and to some extent for bilinear models, whereas applications and implementation of sparsity in connection with tensor decomposition is a less explored area. For details on algorithms see Appendix 1.

2.2. Properties of using the L_1 norm

L_1 norm penalized models have sparse parameters as described in the sections above. However, apart from being sparse, such models exhibit additional properties due to the L_1 norm penalty/constraint.

2.2.1. Not more parameters than the rank

Imagine a linear regression problem $\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e}$ as sketched in Section 1 (Eq. (1)) with $(p > n)$. The aim is to find a solution ($\hat{\mathbf{b}}$) that provides small residuals. Constraining the solution by bounding the L_1 norm of $\mathbf{b}$ ($\|\mathbf{b}\|_1 \le c$) such that the constraint is active, will produce a solution with a maximum of r nonzero elements in $\mathbf{b}$ [5,14], where r is the mathematical rank of $\mathbf{X}$. Hence, when there are few samples in the data set, it can limit the number of variables that can be selected. This may be a problem in practice. Also, in case (of a small subset) of highly correlated columns in $\mathbf{X}$, the selection property of the Lasso results in *one* active (non-zero) variable while the remaining variables are passive (zero) in the model. This is problematic when dealing, for example, with spectral data, where a meaningful solution has the *same* sign and (close to same) magnitude in the parameter space for highly correlated variables.

In regression problems with highly co-linear $\mathbf{X}$-variables, and where the desired number of active variables is not supposed to be limited by the rank of $\mathbf{X}$, there are variants of the Lasso which can cope with these issues, for example the grouped Lasso and elastic net. In grouped Lasso the variables are partitioned into predefined (possibly overlapping) groups. The penalty either kills the entire group or keeps them all [15]. Elastic net proposed by Zou and Hastie [14] extends the Lasso with a ridge penalty. The ridge penalty effectively handles co-linearity while the Lasso penalty works as a variable selector. Compared to the grouped Lasso, there is no need for a predefined grouping of variables. On the other hand, a balance between the L_1 and L_2 penalties must be selected (e.g. by a grid search evaluated by crossvalidation). For further details consult Zou and Hastie [14] and Zhao et al. [15].

2.2.2. Shrinkage of active coefficients

Minimizing the L_1 norm in a regression context leads to variable selection but also implies shrinkage of the *active* coefficients. In case of many irrelevant variables the optimal solution is highly penalized and this can have a significant impact on the magnitude of the active coefficients. A fix to this could be to initially use an L_1 norm penalized solution to determine the sparsity followed by a normal least squares solution to fit the active parameters [16].

2.2.3. Non monotonicity

The Lasso solution path is not monotonic in terms of inclusion or sign. This means that comparing two solutions; one with a high penalty (few active variables) and one with a low penalty (many active variables), the active variable set for the *high penalty* solution is *not* necessarily a subset of the active variable set of the *low penalty* solution. This issue also holds for the sign of the coefficients (see e.g. the two variables (one and five) in example Section 2.1 — Fig. 2, which obtain both positive and negative regression coefficients for different penalties).

2.3. Other features of the L_1 norm

Apart from sparsity, minimizing or bounding of the L_1 norm enjoys other interesting properties.

2.3.1. Compressed sensing

An under-sampled sparse signal can be exactly recovered through an optimization problem including minimization of an L_1 norm under some weak assumptions, and is for example used in medical magnetic resonance image recovery [1,3,17,18].

2.3.2. Robust statistics

Robust statistics is a scientific discipline dealing with methods that return results insensitive to outliers. One of the tricks here is to replace a *sum of squared error* loss function (L_2 norm) with a *sum of absolute error* loss function (L_1 norm). The L_1 norm loss is insensitive to extreme values [19,20]. Note that in the present work the L_1 norm is not used as a loss function but for constraining the model parameters.

2.3.3. Maximum likelihood estimates in case of Laplacian residuals

Assumptions concerning the distribution of residuals are essential in maximum likelihood (ML) fitting of estimates. For example, if the residuals are assumed normally distributed, minimizing the sum of the squared residuals leads to the ML estimates. Likewise, in the case of Laplacian distributed residuals, minimizing the L_1 norm of these returns the ML estimates.

These three interesting and highly relevant subjects related to the use of the L_1 norm are beyond the scope of the present work.

3. Sparse principal component analysis

Sparse principal component analysis aims at estimating a PCA-like model where sparsity is induced on the model parameters; scores and/or loadings. Sparsity in both modes is described by Witten et al. [8] as a special form of penalized matrix decomposition, by Lee et al. [21] as *biclustering* and by Bro et al. [22] as *coclustering*. Sparsity on the loadings only is described in slightly different ways by Zou et al. [7], Witten et al. [8], Jolliffe et al. [6] and Shen and Huang [23]. The following examples are concerned with PCA models with sparsity on the loadings. The abbreviation SPCA will be used for such models from here on.

SPCA on $\mathbf{X}$ $(n \times p)$ with k components can be defined as the solution to the following problem:

$$\underset{\mathbf{T},\mathbf{P}}{\arg\min} \left(\|\mathbf{X} - \mathbf{T}\mathbf{P}^{\mathbf{T}}\|_F^2 \right)$$

subject to $\|\mathbf{p}_i\|_1^1 \le c$ and $\|\mathbf{p}_i\|_2^2 = 1$, for $i = 1, \ldots, k$ where $\mathbf{p}_i$ is the columns of the loading matrix $\mathbf{P}$. $\mathbf{T}$ is the score matrix. c is the L_1 norm constraint on the loading vectors and is the same regardless of the component order.

The PCA solution has both orthogonal scores and loadings. The SPCA criterion does not impose orthogonality between the loading vectors. The components from SPCA are abbreviated SPC. For algorithmic details see Appendix 1.

3.1. Example: jam — sensory and instrumental data

This small example serves to highlight how SPCA is different from ordinary PCA applied to a small data set. Of interest is *how* parameters related to harvest time and location of the raw material (fruits) for jam production are reflected by the quality of the end product.

3.1.1. Data

Twelve samples of jam corresponding to a full factorial design of four locations and three harvest times are evaluated on six instrumental parameters and 13 sensory attributes. For further details see Esbensen [24].

3.1.2. Method

Normal PCA and SPCA are applied to scaled and centered data. Tuning of the sparsity is done such that the number of active variables per component is below 50% (the sum of the absolute elements of the normalized loading vectors should not exceed 2.7 — $\|\mathbf{p}_i\|_1^1 \le 2.7$ for $i = 1, 2$). Orthogonality is not imposed on the scores.

3.1.3. Results

An ordinary PCA solution describes 70% of the total variation in two components (43% and 27% for PC1 and PC2 respectively). The corresponding SPCA solution describes 35% in the first component and 26% in the second component. Due to non-orthogonality, the component-wise variances explained by SPCA are not additive as in ordinary PCA. However, in the present example, the SPCA model describes 61% of the total variation in two components.

The SPCA returns a first component with eight active variables reflecting the *color* of the fruits in connection with *thickness* and *preference*. The second component has a total of nine active variables primarily corresponding to smell and taste attributes (see Fig. 3C and D). The two score vectors are seen to reflect harvest time and location respectively. The score vectors are similar to the scores from ordinary PCA (see Fig. 3A and B). Three variables (*soluble*, *acidity and sweetness*) are discarded (loadings equal to zero) in both components. These variables are among the five least well described variables in the ordinary PCA with less than 21% explained variance in two components. This explains the exclusion.

In order to show how SPCA can deal with irrelevant information, the data set is extended with 100 autoscaled Gaussian random variables. It is of interest to see how similar the results are in terms of scores and loadings. Therefore these are compared between models (PCA and SPCA) on *only* the relevant data (e.g. the two models shown in Fig. 3) and models on data with additional irrelevant variation. For SPCA the sparsity setting is the same for both models ($\|\mathbf{p}_i\|_1^1 \le 2.7$ for $i = 1, 2$). The results are shown in Fig. 4.

It is observed that SPCA is superior in terms of focusing on relevant variables, especially for the first component. For the second component, SPCA only finds three of the nine relevant variables. However,

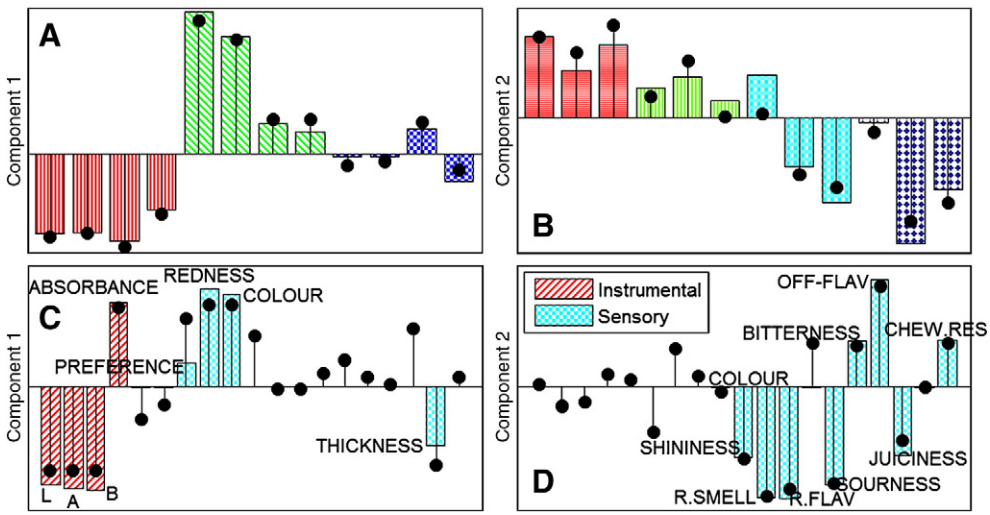

Fig. 3. Results from two component PCA (stems) and sparse PCA (bars) with sparsity constraint on the loadings. A) First score vector from SPCA, with colors corresponding to *location*, B) second score vector from SPCA, with colors corresponding to *harvest time*, C) first loading vector from SPCA, and D) second loading vector from SPCA.

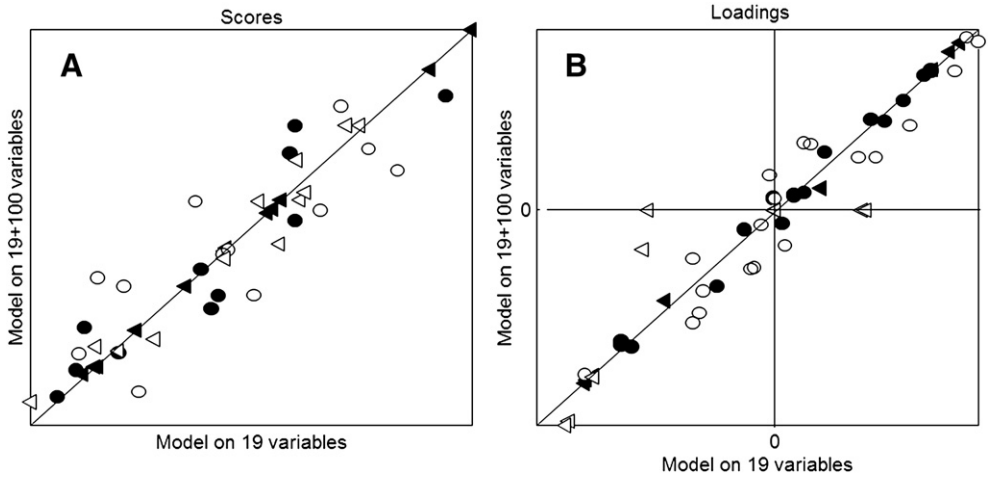

Fig. 4. Comparison of SPCA (◄) and PCA (●) models on data with 19 (relevant) variables and 119 (19 relevant and 100 irrelevant) variables. A: Scores for component 1 (black) and 2 (white). B: Loadings for the *relevant* (19) variables, component 1 (black) and 2 (white). The black line has intercept 0 and slope 1.

the corresponding score vector is quite similar to the one from SPCA on only relevant data (see Fig. 4).

3.2. Example: mass spectrometry data

This example is included to highlight how variable selection can improve model interpretation in cases with several hundreds of features. The serum proteome may reflect the abnormality or pathologic state of organs and tissues. One way to analyze the proteome is through surface enhanced laser desorption ionization time-of-flight (SELDI-TOF) mass spectrometry. This approach has been investigated for prostate cancer diagnostics [25].

3.2.1. Data

Mass spectrometry data of serum proteome was collected from a total of 322 males (age > 50). The 322 samples are partitioned according to the level of serum prostate-specific antigen (PSA) and by *digital rectal exam* and/or *single sextant biopsy* set into 4 groups:

PSA < 1 with no evidence of prostate cancer ($n = 63$)
PSA > 4 with benign tumor ($n = 190$)
4 < PSA < 10 with prostate cancer ($n = 26$)
PSA > 10 with prostate cancer ($n = 43$).

In total 15,154 m/z values are recorded for each sample (for details see Petricoin III et al. [25]).

3.2.2. Method

Initially the data are reduced to 6000 variables by removal of the low m/z region (0–3132) and the high m/z region (12,540–19,996).

The huge amount of variables ($p = 6000$) represents a computational challenge but may potentially also hamper the data analysis as these 6000 variables most likely do not represent 6000 independent chemical analytes. In fact, an SPCA model on these 6000 variables will be heavily weighted towards big peaks merely because of the larger variation of these peaks (results not shown). This rather arbitrary masking of more subtle variations is in essence an unfortunate up-weighting of big (non-informative) peaks. In order to allow all chemical variations to influence the modeling, it is desirable to reduce the number of variables and to represent single peaks in a few *pseudo* variables. Firstly the spectra are manually divided into intervals in such a way, that each interval optimally contains one peak. Some peaks are overlapping, meaning that single intervals can reflect several peaks. Each interval is integrated by performing a singular value decomposition (SVD) on the whole interval. The first component reflects the magnitude of the peak but in case of more complex peaks representing more variation, several components (up to ten) are retained if needed to describe at least 95% of the variation. These normalized score vectors from the SVD across all intervals are held in a matrix **U** and used as a compressed representation of the main variation and used as input data for PCA and SPCA. The procedure is described in detail in Appendix 2.

3.2.3. Bilinear decomposition

U is decomposed by SPCA and PCA in a few latent variables (k) such that

$$\mathbf{U} = \mathbf{T}\mathbf{P}_{\mathbf{U}}^{\mathbf{T}} + \mathbf{E}_{\mathbf{U}} \tag{5}$$

where $\mathbf{T}$ ($n \times k$) is the score matrix, $\mathbf{P_U}$ ($p_U \times k$) the loading matrix and $\mathbf{E_U}$ ($n \times p_U$) the residuals. For SPCA sparsity is imposed on the loadings $\mathbf{P_U}$ ($\|\mathbf{P_{Ui}}\|_1 \le c$ for $i = 1, \ldots, k$, $\mathbf{P_{Ui}}$ is the columns of $\mathbf{P_U}$). The scores in the SPCA model are forced to be mutually orthogonal. This option is, opposite to normal PCA, an active constraint, and can therefore result in a model with lower explained variance. However, this does not lead to significantly different results for the current data, and is included to highlight this option.

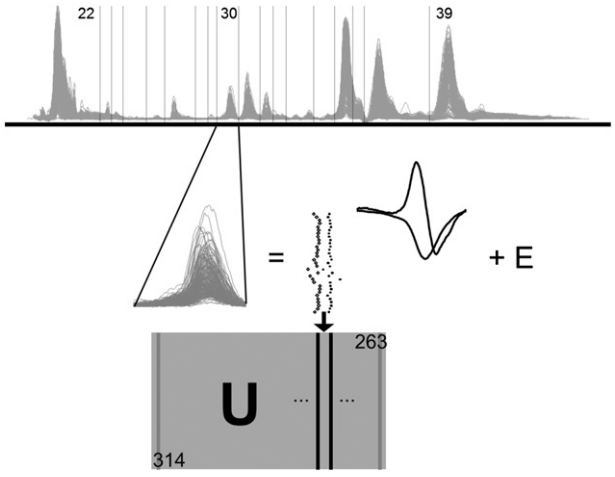

Fig. 5. Display of transformation of data into pseudo variables. The entire spectra are divided into 42 intervals (here interval 22 to 39 are shown). Each of these intervals is mined by SVD on centered data (here shown for interval 30 with two components). The (normalized) scores are collected in a new matrix (**U**) of pseudo variables.

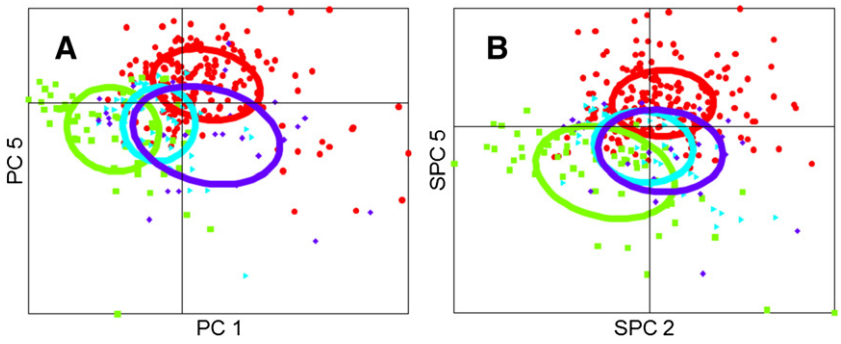

Fig. 6. A: Score plot of PC1 (5.7%) vs. PC5 (2.5%) from the PCA model on truncated data (314×269). B: Score plot of component 2 (SPC2 (3.2%) vs. SPC5 (1.7%)) from the SPCA model on truncated data (314×269). The colors correspond to: PSA<1 with no evidence of prostate cancer (green ■), PSA>4 with benign tumor (red ●), 4<PSA<10 with prostate cancer (purple ♦), and PSA>10 with prostate cancer (turquoise ►). The ellipsoid reflects the subsample distribution centered at the mean and with half axis corresponding to the standard deviation. The ellipsoids are rotated according to subsample correlation between the two components. The explained variance is calculated for the truncated data (322×263).

3.2.4. Results

Initially, eight samples are removed as outliers due to high influence. In total the 6000 variables are partitioned into 42 intervals of varying lengths (20 to 1000). From the 42 intervals, a total of 263 pseudo variables are calculated. In Fig. 5 this process is schematized.

PCA and SPCA with ten components are calculated on these 263 pseudo variables. The constraint in SPCA is chosen such that approximately 30 pseudo variables are active (out of 263). This is achieved by penalizing $||\mathbf{P_{Ui}}||_1 \leq 4$.

The scores are used directly for exploration of grouping in data. The loadings are back transformed into the variable space of **X** by $\mathbf{VP_U}$ ($p \times 10$) for interpretation. **V** is a block diagonal matrix with the interval based SVD loadings (see Appendix 2).

A score plot of PC1 vs. PC5 is shown in Fig. 6A from the PCA model, and in Fig. 6B, SPC2 vs. SPC5 of the SPCA model is shown. From the figures, it seems that there is some grouping in data related to prostate cancer status, and that this grouping can be explored by PCA and/or SPCA.

In order to interpret this separation the corresponding loadings are examined (see Fig. 7).

For SPC2, 19 of the pseudo variables are active (out of 263). This turns out to correspond to 19 intervals (out of 42). Recall that each pseudo variable corresponds to an SVD component. The selected SVD components are primarily the first (#12) but also the second (#6) and a single one forth, reflecting that the interval wise most influential components are also the overall most influential. Similarly for SPC5 37 variables (out of 263) are active representing 25 (out of 42) intervals. The 37 variables are distributed over SVD components (SVDc) as follows: SVDc1 (#14), SVDc2 (#12), SVDc3 (#5), SVDc4 (#3), SVDc5 (#1), SVDc6 (#1) and SVDc7 (#1). Combining SPC2 and SPC5, 48 pseudo variables (out of 263) are active (i.e. 8 in both components), representing 29 intervals. The variance explained by the different models is listed in Table 1. Absolute **U** reflects the variance of the truncated variables (**U**) explained by the model. Absolute **X** reflects the variance explained by the model in the **X** data. Relative **X** describes the average percentage of explained variance by the model across all intervals (for details see Appendix 2). The PCA model obviously has higher variance explained for the matrix **U** (absolute **U**) compared to SPCA. For the two other measures of explained variance there is not a clear separation between SPCA and PCA.

3.3. Discussion

As an exploratory tool, PCA in combination with graphics provides an excellent window into multivariate data. However, in cases with many – possibly irrelevant – variables, interpretation of clustering causes etc. can be complicated. The SPCA approach is here shown to give more interpretable loadings for both examples and with a moderate cost in explained variance (see Table 1). Furthermore, SPCA is to some extent able to focus on the significant inter-variable correlation structure, and thereby *de*select the variables that only have smaller correlations with the remaining ones. This is realized from the results on the *jam* data extended with random variables, and from the mass spectrometry data where in general the first and most significant (interval) SVD components are selected, while the small (and irrelevant) SVD components are deselected. For the mass spectrometry data, entire regions of the spectrum are deselected, and the selected parts seem to represent fewer features than is the case for ordinary PCA. The initial truncation step into pseudo-variables for mass spectrometry data is here applied to obtain *single* variables that mimic *single* analytes as opposed to the raw data which contain highly correlated variables representing the same feature.

4. Sparse inverse covariance matrix

Assessing the correlation structure between variables, e.g. by PCA, helps in understanding data and points towards which variables that exhibit similar patterns. Understanding the pairwise association between two variables in a multivariate model is however not entirely unraveled

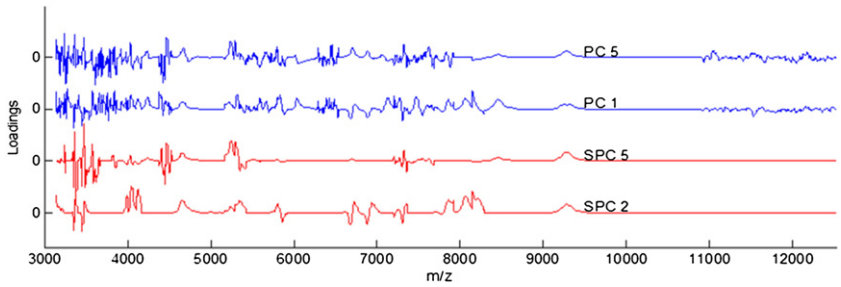

Fig. 7. Loadings from PCA (PC1 and PC5 in blue) and SPCA (SPC2 and SPC5 red) models.

Table 1
Cumulative relative explained variance (%) calculated on the: truncated data (absolute U), taking the individual variables explained variance for the entire **X** into account (absolute **X**) and average over the relative variance captured for each interval (relative **X**).

	Component	1	2	3	4	5
PCA	Absolute **U**	6	11	15	18	21
	Absolute **X**	41	45	63	67	73
	Relative **X**	19	27	36	44	48
SPCA	Absolute **U**	3	6	9	11	13
	Absolute **X**	15	54	56	57	66
	Relative **X**	14	27	36	39	42

by crude correlation patterns. Imagine two correlated variables e.g. *shoe size* and *weight*. A relevant question could be: *Is increase in shoe size directly related to increase in weight, or is this apparent association mitigated through other variables such as height*? This kind of relational structure is reflected by the inverse covariance matrix and can hence be investigated by exploration of this. The covariance matrix keeps information related to pairwise correlation between variables, and the inverse of this reflects partial covariance (correlation)/conditional relation. For details see Appendix 3.

It is important to emphasize that causal inference estimation demands that the data represent *everything* relevant in the context. As this is seldom the case, the presented method can be seen as an explorative, hypothesis generating tool.

By estimation of the inverse covariance matrix from data, information concerning the partial relation between variables is revealed. The least squares estimate of the inverse covariance matrix is however dense, and all variables are therefore more or less partially correlated. Small partial correlations are assumed insignificant, and could be represented by a zero. Friedman et al. [26] formalized the estimation of the inverse covariance matrix as an optimization problem including an L_1 penalty on the single (off diagonal) elements. Solving this problem with feasible penalties results in a sparse solution, i.e. some of the partial correlations (or covariances) being exactly zero and hence independency conditional on all the other variables. The method is called *graphical Lasso*. Formalized, it goes as follows.

Let $\mathbf{X}$ $(n \times p)$ be some normally distributed data structure with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Let $\mathbf{S}$ be the empirical covariance matrix and let $\boldsymbol{\Theta}$ be the inverse covariance matrix ($\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$). The maximum likelihood estimate of $\boldsymbol{\Theta}$ can be found by maximizing the following:

$$\log\det(\boldsymbol{\Theta}) - \mathrm{trace}(\mathbf{S}\boldsymbol{\Theta}) \tag{6}$$

which has the intuitive solution $\hat{\boldsymbol{\Theta}} = \mathbf{S}^{-1}$. Adding an L_1 penalty to the individual *off diagonal* elements of $\boldsymbol{\Theta}$ gives the following maximization problem:

$$\log\det(\boldsymbol{\Theta}) - \mathrm{trace}(\mathbf{S}\boldsymbol{\Theta}) - \lambda \mathrm{P}(\boldsymbol{\Theta}) \tag{7}$$

where $\mathrm{P}(\boldsymbol{\Theta})$ is the sum of the absolute values of the off diagonal elements of $\boldsymbol{\Theta}$, and λ is a positive tuning parameter. Solving Eq. (7) with a feasible penalty (λ) leads to a sparse $\hat{\boldsymbol{\Theta}}$ and hence variables being estimated as conditionally independent.

The graphical Lasso can either be used for exploratory purposes, where solutions to a grid of penalties reveal the strong and weak partial correlations between variables, or by proper tuning of the penalty parameter e.g. by crossvalidation to reveal a single undirected network (graph). Huang et al. [16] used this approach to extract brain connectivity of Alzheimer's disease.

4.1. Example — UN data

Comparison of countries on several specific performance parameters and country characteristics is used for e.g. ranking like the PISA (Programme for International Student Assessment), and plays a crucial role in politics. In this perspective, it is rather important to be able to point to possible indications of seemingly *direct* relations as opposed to *indirect* relations. The current example serves to show the method.

4.1.1. Data

Seven variables (*#inhabitants/physician*, *infant death/1000 inh.*, *#students/100,000 inh.*, *% literate over age of 15 years*, *gross national product* (*GNP*), *population/km²* and *population/1000 ha of agriculture*) are evaluated on a total of 47 countries (two outlying countries (Singapore and Hong Kong) are removed). For further details on data see Gunst and Mason [27].

4.1.2. Method

Initially the data are log- or square root transformed in order to obtain approximate normality. A PCA (centered and scaled) loading plot is used for graphical presentation of the crude correlation structure. Graphical Lasso is used for estimation of the partial correlations and especially where these can meaningfully be considered to be zero. The tuning parameter is optimized by crossvalidation. The partial correlations are indicated on the PCA loading plot as edges. The two models of data (a PCA model and a sparse estimate of the inverse covariance matrix) are estimated independently. It is only in the graphical presentation that the results are combined.

4.1.3. Results

The loading plot of the first two principal components reveal that *#students/100,000 inh.*, *% literate over age of 15 years* and *GNP* are correlated and oppositely correlated to *#inhabitants/physician* and *infant death/1000 inh.* (PC1), while these five variables are not related to *population/km²* and *population/1000 ha of agriculture* which together mainly form PC2. Estimating the inverse correlation matrix reveals which variables in the context are directly related, and which are indirectly related and hence possibly mediated through other variables. In Fig. 8 the edges indicate a *direct* relation (non-zero element of the inverse covariance matrix). *GNP* and *#student/100,000* are for example directly related.

4.1.4. Discussion

Among a variety, two observations are highlighted.

1) There is no edge between *#inhabitants/physician* and *infant death/1000 inh.*

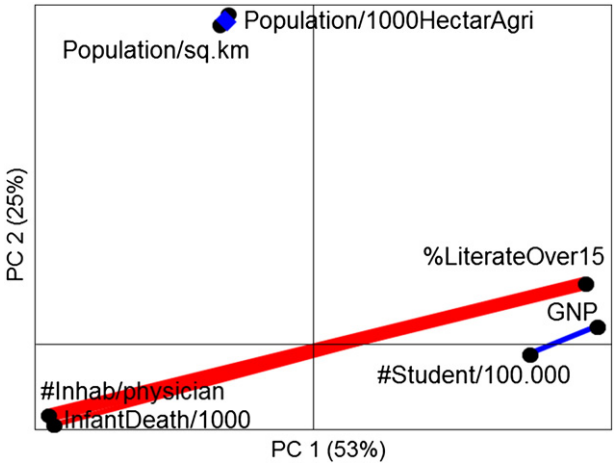

Fig. 8. PCA loading plot of the first two principal components from a PCA on 7 variables across 43 countries. Imposed are edges where the corresponding partial correlation is different from zero. The thickness is proportional to the magnitude and the color reflects the sign (red — negative association, blue — positive association).

2) There are thick edges from *% literate over age of 15 years* to *#inhabitants/physician* and *infant death/1000 inh.* respectively.

This means that infant death is only related to the number of physicians through the adult literate capacity of the country (*% literate over age of 15 years*). Although this pattern is correlated to *GNP* and the number of students, the partial dependency is weak and zero under this penalty. Interpretation of this pattern suggests the hypothesis, that the adult literate capacity is the foundation for both low number of infant deaths and higher number of physicians. However, the partial correlations are without direction and a strictly mathematical and obscure interpretation could just as well be that a low number of infant deaths lead to high adult literate capacity. A limitation of this method is that two correlated, but conditionally independent variables might be connected in the case where the mediating variable is *not* measured or omitted from the analysis. For example, one could speculate that the literate capacity, in relation to the number of physicians, is a surrogate marker for *how much* the government prioritizes health politics. But as this variable is not quantified and included in the analysis, it appears as the two features are directly correlated.

5. Sparse gradient

For many types of data and models, sparsity is not the most obvious constraint on parameters. Under such circumstances imposing sparsity obviously does not improve the model. However sparsity might be valid for an alternative representation of the parameters. Imagine a multivariate process data recorded over time. That could for example be a chemical process or a fermentation monitored by a spectroscopic method reflecting the chemical composition continuously over time. Mechanistic knowledge concerning the process suggests that specific chemical compounds change over time in a systematic fashion, for example a constant concentration for every time point except at a few transition points where the concentration changes, or a linear increase of concentration with a change in slope at a few transition time points. These two concentration profiles have sparsity in the first and second derivatives respectively, with non-zero elements at the transition points. Chemical reactions can sometimes be expressed as exponential decays with an observed transition in reaction characteristics at a certain transition point. Exponential decay of the concentration of an analyte (y) over time (t) can formally be written as:

$$y = y_0 + \exp(-\alpha \cdot t)$$

where α is the reaction constant and y_0 is the concentration at $t=0$. Transition here is the time point where the reaction constant (α) changes. Initially, first order derivation is conducted to handle constant offsets. This is followed by logarithmic transformation to linearize the exponential function. A further derivation leads to a sparse representation where the non-zero point is at the change in reaction constant. Imposing this knowledge in a decomposition of data can reveal the transition points, and the corresponding loadings (e.g. spectra). These three examples are shown in Fig. 9 where a transformation of the original data has sparse properties.

5.1. Example — synthetic data

The data for this example can be interpreted as spectral data reflecting a continuous process over time. One of the analytes is assumed to follow a systematic time pattern. The data ($\mathbf{X}$) are formed with k analytes. For the first analyte there is a piecewise constant response related to time. For the remaining $k-1$ analytes there is no structure related to time. This is clearly an idealized setup, but serves to show how knowledge concerning the system can be incorporated in the data analysis.

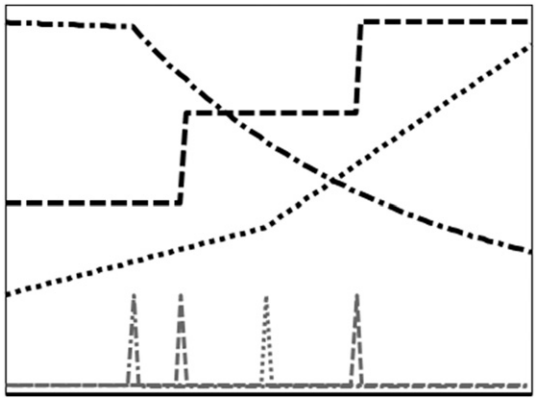

Fig. 9. Three examples of data where sparsity is present for some transformation. — — Three constant functions with different offsets. ▪▪▪ Piece wise linear function with different slopes. — · — Two exponential decays. At the bottom in gray the sparse transformation is shown.

Let $\mathbf{X}$ ($n \times p$) be data formed as $\mathbf{TP}^T$ ($\mathbf{T}$ ($n \times k$) and $\mathbf{P}$ ($p \times k$)), where the first column of $\mathbf{T}$ is piecewise constant (has a sparse gradient), and the remaining $k-1$ columns of $\mathbf{T}$ and the loadings $\mathbf{P}$ are randomly drawn from a Gaussian distribution. The knowledge concerning sparsity on the gradient is imposed in estimation of the first component ($\mathbf{t}$ and $\mathbf{p}$) such that the following optimization problem is solved.

$$\arg\min_{\mathbf{t},\mathbf{p}} \left(\|\mathbf{X} - \mathbf{t}\mathbf{p}^T\|_F^2 \right) \text{ subject to } \sum_{i=2,\ldots,p} \|t_i - t_{i-1}\|_1 \leq c$$

This step is built into the NIPALS algorithm for calculation of $\mathbf{t}$, when $\mathbf{p}$ is known (see Appendix 1). The tuning parameter c is selected in accordance with the assumption concerning piecewise constant response in such a way that the shape of the score vector shows distinct constant regions. In practice, models with different tuning values are compared and the best model is chosen subjectively.

5.1.1. Results

In Fig. 10 the results are shown for $n=100$, $p=50$ and $k=4$. The sparse component contributes with 17% of the total variance. Compared with an ordinary four component PCA, the component with the highest correlation with the *true* loading was 0.62 (0.99 for the component recovered by L_1 constrained PCA, see Fig. 10B). Furthermore the corresponding score vector of the ordinary PCA model, though increasing, the shape is not a clear step function, and estimation of transition points is hence difficult (see Fig. 10A).

5.1.2. Discussion

These data are fairly idealized, but the example still highlights the usefulness of being able to impose e.g. a step function in the data. It is of utmost importance to justify such constraints based on mechanistic knowledge concerning the system. The example shows that imposing sparsity when it is known to be valid can help in unraveling the process that actually generated the data.

6. Software

PCA is conducted using PLS_Toolbox ver. 6.0.1 for Matlab® R2010b ver. 7.11.0.584. SPCA is conducted using the SPCA algorithm available from http://www.models.life.ku.dk/sparsity — December, 13th 2011.

Baseline correction is done using the msbackadj function from the bioinformatics toolbox (ver. 3.6) for Matlab® (R2010b). Horizontal alignment is conducted using the *i*coshift function for Matlab® with

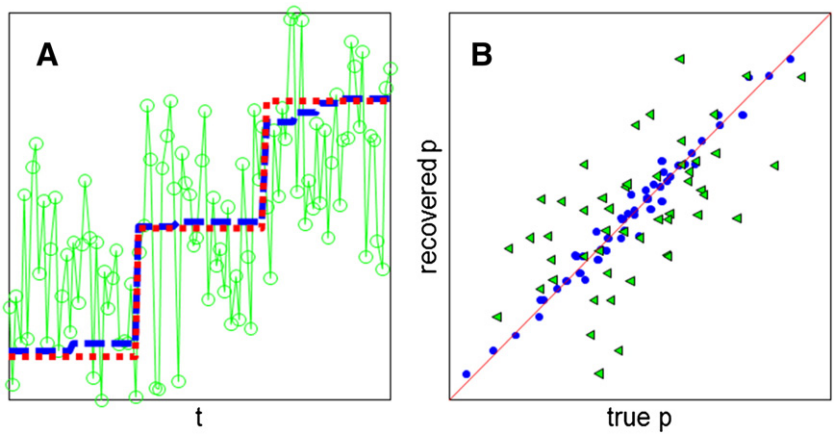

Fig. 10. Results from estimation of the first score and loading from data with rank four, where estimation of the first component is with an L_1-norm constraint on the gradient of the score vector. For comparison the *best* (highest correlation) PCA component, from a four component model is imposed. A: True (red), recovered (blue) and PC3 (green) score vectors. B: Comparison of the corresponding loading vectors. Recovered by L_1 constraint model (blue ●) and PC3 (green ◄).

three windows for each interval (available from http://www.models.life.ku.dk — December, 13th 2011).

Graphical Lasso is performed using glasso for R (ver. 2.11.1).

7. Discussion and conclusion

Presented here are methods mostly developed in the signal processing and statistical communities. These methods can beneficially be translated into a chemometric framework. Sparsity can be achieved through an L_1 norm penalty/constraint, and hence add an additional tuning parameter on top of selecting proper preprocessing, number of components, etc. We have shown a few applications, where existing models are modified via an L_1 norm constraint. Sparse principal component analysis (SPCA) estimates a model with only a subset of the variables being active in each component. This has the potential advantage of being easier to interpret compared to ordinary PCA, where all variables appear in all components. PCA is, as an exploratory tool, often used to examine the correlation structure between variables. Likewise we here use the graphical Lasso to explore the partial correlation structure between variables, and especially which variables can be considered conditionally independent. The graphical Lasso in combination with PCA correlation patterns provides a powerful tool for data exploration and hypothesis generation. Imposing *relevant* prior system knowledge in the model building leads to more powerful models. If it is known, that for example the parameters – or a transformation of those – are truly sparse, this knowledge can be imposed via an L_1 norm penalty. We show, that utilizing knowledge concerning sparsity in a bilinear decomposition of an idealized data set, uncovers the components that generated data. Sparsity is a useful add-on to models from the chemometric toolbox, and it is hoped that this tutorial can facilitate further exploration of this area.

Acknowledgment

Professor Robert Tibshirani is acknowledged for inspiration.

Appendix 1. Algorithms

Although an L_1 penalized least squares problem is often convex (or bi-convex), there do not exist analytical closed form solutions. Off the shelf convex optimization solvers such as cvx are easy to use [28], e.g. solving the Lasso for a fixed penalty using cvx requires just four lines of Matlab code. Alternatively, the solver can be built into a larger algorithm, and an example of such is given below (see Algorithm for fitting "PCA-like" model with a sparse gradient on the scores below). A drawback of using general purpose solvers is lack of speed. More targeted algorithms have been developed for many specific types of problems.

Regression

Efron et al. [29] developed the least angle regression (LARS) algorithm which can estimate the entire Lasso solution path for continues response in an order of a least squares fit. The algorithm is implemented in the R package lars. For variants of regression problems; logistic-, probit-, Poisson-, multinomial- and Cox regression, where the response is modified through a link function, Friedman et al. [30] developed an extremely fast and less restricted algorithm based on *cyclic coordinate descend* which fits a path of solutions corresponding to a grid of penalties. This algorithm (glmnet) is implemented in the R package glmnet and is also available for Matlab® (http://www-stat.stanford.edu/~tibs/glmnet-matlab/ — December, 13th 2011).

Bilinear models

There exist several approaches for obtaining a bilinear model with sparsity imposed on the scores or/and the loadings. Here three slightly different methods are presented.

The algorithm proposed by Zou et al. [7] (SPCA), that imposes sparsity in *one* direction, relies on the fact that the PCA scores are linear combinations of $\mathbf{X}$ and hence can be formulated as a regression problem and estimated by the Lasso in order to achieve sparse loadings.

The sparse principal components (SPC) procedure by Witten et al. [8] is formulated as a bi-convex optimization problem for estimation of a single component (normalized-scores ($\mathbf{u}$) and loadings ($\mathbf{v}$)).

$$\arg\max_{\mathbf{v}} \left(\mathbf{u}^T\mathbf{X}\mathbf{v}\right) \text{ subject to } \|\mathbf{v}\|_2 \le 1, \|\mathbf{u}\|_2 \le 1, \mathbf{v}_1 \le c \tag{A1}$$

The method iteratively fixes one mode (e.g. scores) and estimates the other mode (e.g. loadings) until convergence. Estimation of vectors with L_1 constraint (e.g. loadings in Eq. (A1)) is conducted via soft thresholding of the least squares estimates (see below). In case of multiple components, this method works with deflation of $\mathbf{X}$ and estimation of consecutive components based on the current residuals, leading to a solution nested in the components.

In the present work we have used an algorithm that estimates an SPCA solution for a fixed number of components (k) as the solution to:

$$\arg\min_{\mathbf{T},\mathbf{P}} \left(||\mathbf{X}-\mathbf{T}\mathbf{P}^{\mathbf{T}}||_F^2 \right) \quad \text{(A2)}$$

subject to $||\mathbf{p}_i||_1^1 \le c$ and $||\mathbf{p}_i||_2^2 = 1$, for $i = 1, \ldots, k$ where $\mathbf{p}_i$ is the columns of the loading matrix $\mathbf{P}$ and c is the L_1 norm constraint on the loading vectors. The method estimates the score matrix ($\mathbf{T}$) based on the current loadings: $\mathbf{T} = \mathbf{X}\mathbf{P}^+$ (where $\mathbf{P}^+$ is the pseudo inverse of $\mathbf{P}$). The loading matrix ($\mathbf{P}$) is estimated based on the current score matrix ($\mathbf{T}$), and is a column-wise soft thresholded version of the least squares estimates $\mathbf{p}_i = S(\mathbf{p}_{iLS}, \lambda_i)$, where $\mathbf{p}_{iLS}$ is the column-wise least squares estimate and $S(x,\lambda)$ is the soft threshold operator working on the individual elements:

$$S(x,\lambda) = \begin{cases} x+\lambda & \text{for} \quad x < -\lambda \\ 0 & \text{for} \quad -\lambda \le x \le \lambda \\ x-\lambda & \text{for} \quad x > \lambda \end{cases}.$$

The tuning parameter λ_i is chosen such that the L_1 (and L_2) norm constraint is fulfilled for each component. The algorithm iterates between estimation of the score and loading matrix until convergence. Due to similarities with alternating least squares (ALS) the name alternating shrunken least squares (ASLS) is suggested. For one component models the solutions to Eqs. (A1) and (A2) are identical.

None of the approaches are strictly convex, and a solution might therefore be a local minimum.

SPCA by Zou et al. [7] is implemented in the R package elastic net and is available for Matlab® (http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=3897 — December, 13th 2011). The SPC algorithm by Witten et al. [6] for sparsity imposed in *one* or *two* directions is implemented in the R package PMA. Algorithms for component wise estimation with deflation (similar to the SPC algorithm by Witten et al. [8]) and ASLS are available on http://www.models.life.ku.dk/sparsity (December, 13th 2011).

Multiway models

Multiway algorithms for decomposition of higher order arrays (tensors) can be modified by addition of a penalty to inner algorithmic least squares criterions. Martínez-Montes et al. [31], Mørup et al. [32], Papalexakis et al. [33] and Allen [34] are examples of algorithms for achieving sparsity in one (or several) mode(s) of the PARAFAC and Tucker models. The algorithms are as far as the authors know not made publicly available.

Algorithm for fitting "PCA-like" model with a sparse gradient on the scores

This is a modified NIPALS (non linear iterative partial least squares) algorithm [35] for estimation of a PCA like solution with a sparse gradient on the scores. Here step 3 (the update of the score vector) is extended with an L_1 norm penalty on the gradient of the score vector.

The gradient ($\partial\mathbf{t}$) of a numerical vector ($\mathbf{t}$) ($n \times 1$) can be calculated as $\partial t_i/\partial i = t_i - t_{i-1}$. In matrix notation this is $\partial\mathbf{t} = \mathbf{t}^{\mathrm{T}}\mathbf{M}$, where $\mathbf{M}$ ($n \times n-1$) is a bi diagonal matrix[1] with -1 in the main diagonal and 1 in the lower bi diagonal (i.e. $\mathbf{M}_{j,j} = -1$ for $j = 1, \ldots, n-1$ and $\mathbf{M}_{j,j-1} = 1$ for $j = 2, \ldots, n$). The algorithm for extraction of one component with an additional L_1 penalty is as follows:

0. Initialize $\mathbf{t}$ (e.g. first score vector from normal PCA, a column of $\mathbf{X}$, random numbers …).
1. Calculate $\mathbf{p}$: $\mathbf{p} = \mathbf{X}^{\mathbf{T}}\mathbf{t}$.
2. Normalize $\mathbf{p}$: $\mathbf{p} = \mathbf{p}/||\mathbf{p}||_2$.
3. Calculate $\mathbf{t}$:

$$\arg\min_{\mathbf{t}} \left(||\mathbf{X}-\mathbf{t}\mathbf{p}^{\mathbf{T}}||_F^2 \right)$$

subject to

$$||\mathbf{t}^{\mathbf{T}}\mathbf{M}||_1 \le c.$$

4. Repeat 1–3 until convergence.

$||\cdot||_F^2$ is the Frobenius norm of the matrix (sum of squared elements). Step 3 is in the current work performed using the cvx solver [28].

[1] $\mathbf{M}$ is not exactly bi diagonal as it is not symmetric.

Appendix 2. Interval wise data truncation via singular value decomposition (SVD)

Data are initially baseline corrected and centered.
Divide the mass spectra into I intervals based on spectral examination.
For each interval $i = 1, \ldots, I$ of $\mathbf{X} = [\mathbf{X}_1\mathbf{X}_2 \cdots \mathbf{X}_I]$ do:

Horizontal alignment of each spectrum.
Singular value decomposition (SVD) $\mathbf{X}_i = \mathbf{U}_i\mathbf{S}_i\mathbf{V}_i^{\mathbf{T}}$, where $\mathbf{U}_i$ and $\mathbf{V}_i$ represent the left- and right eigenvectors of $\mathbf{X}_i$ respectively, and $\mathbf{S}_i$ is diagonal with the singular values of $\mathbf{X}_i$.
The number of components is selected such that at least 95% of the variance in $\mathbf{X}_i$ is explained, albeit a maximum of 10 components.
The truncated left eigenvectors ($\mathbf{U}_i^*$) are stored as condensed pseudo variables $\mathbf{U} = [\mathbf{U}_1^*\mathbf{U}_2^* \cdots \mathbf{U}_I^*]$ ($n \times p_U$) representing the main variation of $\mathbf{X}$.
The truncated right eigenvectors ($\mathbf{V}_i^*$) are stored as a large *block diagonal* loading matrix

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1^* & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_2^* & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{V}_I^* \end{bmatrix} (p \times p_U).$$

Variance explained

Variance explained can be calculated at several different levels.
Variance explained for the decomposition of $\mathbf{U}$ (Eq. (5)) can be calculated directly as:

$$EV_{\text{absolute}\,\mathbf{U}} = 1 - tr\left(\mathbf{E}_{\mathbf{U}}^{\mathbf{T}}\mathbf{E}_{\mathbf{U}}\right)/tr(\mathbf{U}_{\mathbf{T}}\mathbf{U}).$$

Each variable/column of $\mathbf{U}$ explains a certain percentage of the variance in $\mathbf{X}$ (corresponding to the (squared) singular values from the interval SVD). This can be taken into account to reveal an *overall* variance explained in $\mathbf{X}$. Let $\mathbf{s}^2$ ($p_U \times 1$) be a vector of squared singular values corresponding to the columns of $\mathbf{U}$, and let $\mathbf{w}$ ($p_U \times 1$) be a vector of the variance explained (in percentage) for each column in $\mathbf{U}$ by the (S) PCA model. Then the overall variance explained in $\mathbf{X}$ ($EV_{\text{absolute}\,\mathbf{X}}$) can be calculated as:

$$EV_{\text{absolute}\,\mathbf{X}} = \mathbf{w}^{\mathbf{T}}\mathbf{s}^{\mathbf{2}}/\mathbf{tr}\left(\mathbf{X}^*\mathbf{X}^{*\mathbf{T}}\right)$$

where $\mathbf{X}^*$ is the data which have been baseline corrected, shifted and centered interval wise as described above.

Some intervals contribute more than others, simply due to magnitude of single peaks. The overall variance explained ($EV_{\text{absolute}\,\mathbf{X}}$) might reflect that single large peaks are well modeled while smaller ones are not. The initial partitioning into intervals is exactly to cope with such issues. Hence the *relative* variance explained for each interval or the mean across all intervals might be a more feasible

measure. Let $\mathbf{s}_i^2$, $\mathbf{w}_i$ and $\mathbf{X}_i^*$ be subvectors/matrices of $\mathbf{s}^2$, $\mathbf{w}$ and $\mathbf{X}^*$, respectively corresponding to the ith interval ($i = 1, \ldots, I$). Then the relative variance explained in $\mathbf{X}$ ($EV_{\text{relative }\mathbf{X}}$) can be calculated as:

$$EV_{\text{relative}\,\mathbf{X}} = \frac{1}{I}\sum_{i=1,\ldots,I} \mathbf{w}_i^{\mathbf{T}}\mathbf{s}_i^2 / \mathbf{tr}\left(\mathbf{X}_i^*\mathbf{X}_i^{*\mathbf{T}}\right).$$

Software details are given in Section 6.

Appendix 3. Inverse covariance matrix — partial correlation

Imagine three random variables **x**, **y**, and **z**. Then partial covariance/correlation between **y** and **z** given **x** can intuitively be calculated by regressing **y** on **x** and **z** on **x** followed by calculation of the covariance (correlation) between the two sets of residuals. More formally: let $\rho_{\mathbf{y},\mathbf{z}|\mathbf{x}}$ be the conditional covariance between **y** and **z** given **x**, and $\mathbf{y}(\mathbf{x})$ and $\mathbf{z}(\mathbf{x})$ be the predictions of **y** and **z** given **x**. For centered data:

$$\mathbf{y}(\mathbf{x}) = \operatorname{cov}(\mathbf{x},\mathbf{y})\operatorname{var}(\mathbf{x})^{-1}\mathbf{x}$$

and

$$\mathbf{z}(\mathbf{x}) = \operatorname{cov}(\mathbf{x},\mathbf{z})\operatorname{var}(\mathbf{x})^{-1}\mathbf{x}.$$

Then

$$\begin{aligned}\rho_{\mathbf{y},\mathbf{z}|\mathbf{x}} &= \operatorname{cov}(\mathbf{y}-\mathbf{y}(\mathbf{x}),\mathbf{z}-\mathbf{z}(\mathbf{x}))\\ &= \operatorname{cov}(\mathbf{y},\mathbf{z}) - \operatorname{cov}(\mathbf{y},\mathbf{x})\operatorname{var}(\mathbf{x})^{-1}\operatorname{cov}(\mathbf{z},\mathbf{x}).\end{aligned}$$

Let **S** be the covariance matrix of [**x**,**y**,**z**].

$$\mathbf{S} = \begin{pmatrix} \operatorname{var}(\mathbf{x}) & \operatorname{cov}(\mathbf{y},\mathbf{x}) & \operatorname{cov}(\mathbf{z},\mathbf{x}) \\ \operatorname{cov}(\mathbf{x},\mathbf{y}) & \operatorname{var}(\mathbf{y}) & \operatorname{cov}(\mathbf{z},\mathbf{y}) \\ \operatorname{cov}(\mathbf{x},\mathbf{z}) & \operatorname{cov}(\mathbf{y},\mathbf{z}) & \operatorname{var}(\mathbf{z}) \end{pmatrix}$$

and $\mathbf{S}^{-1}$ the inverse of **S**. From matrix inversion the element corresponding to **y** and **z** (second column, third row or third column second row) has the form:

$$\mathbf{S}^{-1}_{\mathbf{y},\mathbf{z}} = \frac{1}{\det(\mathbf{S})}\left(\operatorname{cov}(\mathbf{x},\mathbf{z})\operatorname{cov}(\mathbf{x},\mathbf{y}) - \operatorname{var}(\mathbf{x})\operatorname{cov}(\mathbf{y},\mathbf{z})\right).$$

From this we get:

$$\rho_{\mathbf{y},\mathbf{z}|\mathbf{x}} = -\det(\mathbf{S})\operatorname{var}(\mathbf{x})^{-1}\mathbf{S}^{-1}_{\mathbf{y},\mathbf{z}} = k\mathbf{S}^{-1}_{\mathbf{y},\mathbf{z}}$$

where k is a scalar constant [36].

References

[1] D. Donoho, Compressed sensing, Technical Report, Department of Statistics, Stanford University, Stanford, 2004.
[2] S. Chen, D. Donoho, in: Basis Pursuit, Signals, Systems and Computers, 1994 Conference Record of the Twenty-Eighth Asilomar Conference on, 1, 1994, pp. 41–44.
[3] M. Lustig, D. Donoho, J.M. Pauly, Sparse MRI: the application of compressed sensing for rapid MR imaging, Magnetic Resonance in Medicine 58 (6) (2007) 1182–1195.
[4] R. Tibshirani, Regression shrinkage and selection via the lasso: a retrospective, Journal of the Royal Statistical Society: Series B 73 (3) (2011) 273–282.
[5] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society: Series B 58 (1996) 267–288.
[6] I.T. Jolliffe, N.T. Trendafilov, M. Uddin, A modified principal component technique based on the LASSO, Journal of Computational and Graphical Statistics 12 (3) (2003) 531–547.
[7] H. Zou, T. Hastie, R. Tibshirani, Sparse principal component analysis, Journal of Computational and Graphical Statistics 15 (2) (2006) 265–286.
[8] D.M. Witten, R. Tibshirani, T. Hastie, A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis, Biostatistics 10 (3) (2009) 515–534.
[9] H. Chun, S. Keleş, Sparse partial least squares regression for simultaneous dimension reduction and variable selection, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72 (2010) 3–25.
[10] M.C. Wu, L. Zhang, Z. Wang, D.C. Christiani, X. Lin, Sparse linear discriminant analysis for simultaneous testing for the significance of a gene set/pathway and gene selection, Bioinformatics 25 (9) (2009) 1145–1151.
[11] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, K. Knight, Sparsity and smoothness via the fused lasso, Journal of the Royal Statistical Society: Series B 67 (1) (2005) 91–108.
[12] J. Bi, K. Bennett, M. Embrechts, C. Breneman, M. Song, Dimensionality reduction via sparse support vector machines, Journal of Machine Learning Research 3 (2003) 1229–1243.
[13] A.E. Hoerl, R.W. Kennard, Ridge regression: biased estimation for nonorthogonal problems, Technometrics 12 (1970) 55–67.
[14] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, Journal of the Royal Statistical Society: Series B 67 (2) (2005) 301–320.
[15] P. Zhao, G. Rocha, B. Yu, Grouped and hierarchical model selection through composite absolute penalties, The Annals of Statistics 37 (6A) (2009) 3468–3497.
[16] S. Huang, J. Li, L. Sun, J. Ye, A. Fleisher, T. Wu, K. Chen, E. Reiman, Learning brain connectivity of Alzheimer's disease by sparse inverse covariance estimation, NeuroImage 50 (3) (2010) 935–949.
[17] E.J. Candès, T. Tao, Decoding by linear programming, IEEE Transactions on Information Theory 51 (2005) 4203–4215.
[18] E.J. Candès, The restricted isometry property and its implications for compressed sensing, Compte Rendus de l'Academie des Sciences, Paris, Serie I 346 (2008) 589–592.
[19] S.A. Vorobyov, Y. Rong, N.D. Sidiropoulos, A.B. Gershman, Robust iterative fitting of multilinear models, signal processing, IEEE Transactions on Signal Processing 53 (8) (2005) 2678–2689.
[20] E.J. Candes, X. Li, Y. Ma, J. Wright, Robust principal component analysis? Arxiv preprint arXiv (2009) (0912.3599).
[21] M. Lee, H. Shen, J.Z. Huang, J.S. Marron, Biclustering via sparse singular value decomposition, Biometrics 66 (2010) 1087–1095.
[22] R. Bro, E.E. Papalexakis, E. Acar, N.D. Sidiropoulos, Coclustering — a useful tool for chemometrics. Submitted for, Journal of Chemometrics 26 (6) (2012) 256–263.
[23] H. Shen, J.Z. Huang, Sparse principal component analysis via regularized low rank matrix approximation, Journal of Multivariate Analysis 99 (6) (2008) 1015–1034.
[24] K.H. Esbensen, Multivariate Data Analysis in Practice, 5th edition Camo Process AS, 2001.
[25] E.F. Petricoin, A.M. Ardekani, B.A. Hitt, P.J. Levine, V.A. Fusaro, S.M. Steinberg, G.B. Mills, C. Simone, D.A. Fishman, E.C. Kohn, L.A. Liotta, Use of proteomic patterns in serum to identify ovarian cancer, The Lancet 359 (9306) (2002) 572–577.
[26] J. Friedman, T. Hastie, R. Tibshirani, Sparse inverse covariance estimation with the graphical lasso, Biostatistics 9 (3) (2008) 432–441.
[27] R.F. Gunst, R.L. Mason, in: Regression Analysis and Its Applications: a Data-oriented approach, Marcel Dekker, NY, 1980, p. 358.
[28] M. Grant, S. Boyd, CVX: Matlab Software for Disciplined Convex Programming, Version 1.21, http://cvxr.com/cvxApril 2011.
[29] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, The Annals of Statistics 32 (2) (2004) 407–499.
[30] J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent, Journal of Statistical Software 33 (1) (2010) 1–22.
[31] E. Martínez-Montes, J.M. Sánchez-Bornot, P.A. Valdés-Sosa, Penalized PARAFAC analysis of spontaneous EEG recordings, Statistica Sinica 18 (2008) 1449–1464.
[32] M. Mørup, L.K. Hansen, S.M. Arnfred, Algorithms for sparse nonnegative Tucker decompositions, Neural Computation 20 (8) (2008) 2112–2131.
[33] E.E. Papalexakis, N.D. Sidiropoulos, R. Bro, From K-means to higher-way co-clustering: multilinear decomposition with sparse latent factors, IEEE Transactions on Signal Processing — Submitted (2011), http://dx.doi.org/10.1109/TSP.2012.2225052.
[34] G.I. Allen, in: Sparse higher-order principal component analysis, Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS), 22, 2012.
[35] H. Wold, Nonlinear estimation by partial least squares procedures, in: Research Papers in Statistics, Wiley, New York, 1966, pp. 414–444.
[36] J. Whittaker, Graphical Models in Applied Multivariate Statistics, John Wiley, Chichester, 1990.