Applications of semidefinite programming in chemometrics

Ray Pörn* and Tom Lillhonga

Novia University of Applied Sciences, Wolffskavägen 33, 65200 Vasa, Finland
*Corresponding author: ray.poern@novia.fi

Introduction

Different problems arising in chemometrics are studied along with their solutions using semidefinite programming (SDP). SDP is a technique for minimising convex functions over a set of positive semidefinite matrices[1]. The problem of finding sparse principal components for a matrix can be modeled as an SDP problem and solved efficiently for a pre-specified target sparsity value[2]. SDP can also be used for optimal distance metric learning[3] and for the construction of quadratic decision boundaries in classification[1,4].

Materials and Methods

In this section the semidefinite optimisation is presented and illustrated using some small examples. Then we proceed to applications in classification and to the computation of sparse loadings in principal components analysis (PCA). The classification procedure is applied to small artificial examples and the sparse PCA is applied on large scale spectroscopy and imaging data.

Semidefinite programming – a short overview

The classical form of an optimisation problem has variables stored in a vector $x \in R^n$. Let A be a matrix with mean-centered columns and covariance matrix $C=A^TA$. The problem of finding the first principal component of A is equivalent to solving the optimisation problem

$$\begin{aligned} \max \quad & x^T C x \\ & x^T x = 1 \end{aligned} \tag{1}$$

The objective function and the constraint are quadratic functions in the variable x. We now define a matrix variable X that models the product of two variables x as $X_{ij} = x_i x_j$ (there are n^2 variables in X). Given this identity, the quadratic function can be linearised in the higher-dimensional space defined by X. We obtain

$$x^T C x = \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij} x_i x_j = \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij} X_{ij} = \text{trace}(CX)$$

where trace is a common notation for a linear function of a matrix variable X. The matrix X is now defined to be a rank-1 matrix according to $X_{ij} = x_i x_j$, *i.e.* $X = xx^T$. Problem (1) can now be stated as

$$\begin{aligned} \max \quad & \text{trace}(CX) \\ & \text{trace}(X) = 1 \\ & X = xx^T \end{aligned} \tag{2}$$

The objective and the first constraint are now linear in X and the second constraint requires X to be a (positive semidefinite) rank-1 matrix. The rank-1 constraint can be relaxed to $X \succeq 0$ (X is a positive semidefinite matrix, *i.e.* all eigenvalues are non-negative). The semidefinite variant of (1) can then be expressed as

$$\begin{aligned} \max \quad & \text{trace}(CX) \\ & \text{trace}(X) = 1 \\ & X \succeq 0 \end{aligned} \tag{3}$$

The last constraint defines the set of positive semidefinite matrices which is a convex set (it is a so-called cone). Problem (3) is a linear SDP. Steps (1)-(3) are common when deriving a semidefinite relaxation of a

Reference paper as:
Pörn, R. and Lillhonga, T. (2012). Applications of semidefinite programming in chemometrics, in:
Proceedings of the 15th International Conference on Near Infrared Spectroscopy,
Edited by M. Manley, C.M. McGoverin, D.B. Thomas and G. Downey, Cape Town, South Africa, pp. 81-86.

non-convex optimisation problem. In this case the relaxation is tight (*i.e.* identical to the original problem). Convex SDPs are solved efficiently using polynomial-time interior point methods[1]. This section ends with another example that models as an SDP problem.

Example: The matrix completion problem.
Given a covariance matrix with unknown element (x) at position (2,3). Let the matrix be

$$C = \begin{pmatrix} 4 & 2.4 & 1.4 & 3 \\ 2.4 & 7 & x & 4 \\ 1.4 & x & 5 & 2.2 \\ 3 & 4 & 2.2 & 4 \end{pmatrix}$$

The problem is now to determine the maximal and minimal values that the covariance x between variables 2 and 3 can attain. Since the covariance matrix is positive (semidefinite by construction), this problem can be solved as the two separate SDPs below

$$\begin{aligned} &\max / \min \quad x \\ &\qquad C \succeq 0 \end{aligned}$$

The optimums are max=5.52 and min= -0.95. So if $x \in [-0.95, 5.52]$ then C is a valid covariance matrix.

Quadratic classification

In pattern recognition and classification problems we are given two (or more) sets of points in n-dimensional space, $\{x_1,...,x_N\}$ and $\{y_1,...,y_M\}$, and we wish to find a function f that is positive in the first set and negative in the second. If such a separation is achieved, the function f perfectly classifies the points. If perfect classification is impossible, we usually seek a function that approximately classifies the points, for example one that minimises the number of misclassified points. This is a hard combinatorial problem and it is often relaxed to the standard (linear) support vector classifier for the sets $\{x_1,...,x_N\}$ and $\{y_1,...,y_M\}$.

$$\text{(LSV)} \qquad \begin{aligned} \min \quad & \|a\|_2 + \gamma(e^T u + e^T v) \\ & a^T x_i - b \geq 1 - u_i, \quad i = 1,..,N \\ & a^T y_i - b \leq -1 + v_i, \quad i = 1,..,M \\ & u \geq 0, v \geq 0 \\ & a \in R^n, b \in R \end{aligned}$$

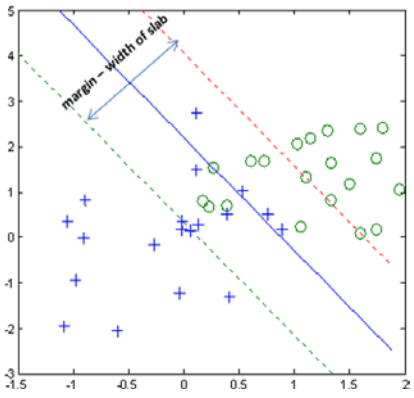

Figure 1. Illustration of a linear support vector on artificial data (γ=0.5).

where e is an all-one vector with appropriate dimensions. The additional slack variables u and v model the amount of violation of points x and y. If u=0 then all x are correctly classified and if v=0 then all y are classified correctly. The width of the margin is $2/\|a\|_2$. The positive trade-off parameter γ gives the relative weight between the number of misclassified points (that we want to minimise) compared to the width of the separating slab (that we want to maximise). This problem is convex and the global optimum is obtained efficiently[1,4]. A simple classification is illustrated in Figure 1 (γ=0.5). In this case, seven points are misclassified and 14 lie within the slab. The linear support vector can be generalised to more complicated functional forms using a kernel function that projects the data into a higher dimensional space where linear separation is possible[4]. The solution to such a problem is called the least-squares support vector machine classifier (LS-SVM). The LS-SVM approach always involves parameter tuning to obtain a well-behaved kernel function and to avoid over-fitting. For a proper choice of the kernel function, the LS-SVM problem is convex.

Using SDP we can construct a quadratic classification scheme that can be interpreted as an intermediate between LSV and LS-SVM. We introduce a quadratic function $f(x) = x^T P x + q^T x + r$ where P is a

Reference paper as:
Pörn, R. and Lillhonga, T. (2012). Applications of semidefinite programming in chemometrics, in:
Proceedings of the 15th International Conference on Near Infrared Spectroscopy,
Edited by M. Manley, C.M. McGoverin, D.B. Thomas and G. Downey, Cape Town, South Africa, pp. 81-86.

symmetric matrix, q a vector and r a scalar. Using similar ideas and variables as in the (LSV) formulation, quadratic classification (QC) of two sets can be modelled as

$$
\text{(QC)} \qquad
\begin{aligned}
\min \quad & \|P\|_2 + \|q\|_2 + \gamma(e^T u + e^T v) \\
& x_i^T P x_i + q^T x_i + r \geq 1 - u_i, \qquad i = 1,..,N \\
& y_i^T P y_i + q^T y_i + r \leq -1 + v_i, \qquad i = 1,..,M \\
& u \geq 0, v \geq 0 \\
& P \succeq 0, q \in R^n, r \in R
\end{aligned}
$$

where $\|P\|_2 = \lambda_{\max}(P)$ (the largest eigenvalue of P) and $\|q\|_2 = \sqrt{q^T q}$. If P=0 (null matrix) then QC collapses to ordinary LSV. The matrix variable P is constrained to be positive semidefinite. The quadratic function defines therefore an ellipsoid that, in an ideal case, encloses the points y. This SDP problem has a convex objective function and linear constraints in the matrix variable P, vector variable q and scalar variable r.

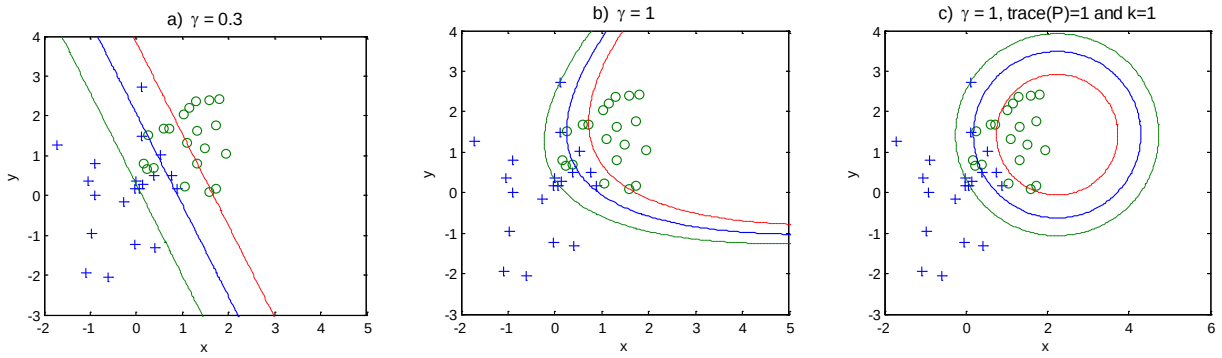

Figure 2. Illustration of quadratic classification. a) γ=0.3 leads to linear classification, b) γ=1 leads to a quadratic surface and c) γ=1, trace(P)=1 and k=1 leads to pure spherical classification.

Some additional constraints that control the shape of the quadratic form can be included in (QC). The matrix P can be regularised by the constraint trace (P)=1; this constraint keeps the matrix P unique. In addition it is known that the function $\lambda_{\max}(P)$ is convex and the function $\lambda_{\min}(P)$ is concave with respect to the matrix variable P. This can, for example, be used to model an upper bound on the eccentricity of the ellipsoid by considering the constraint

$$\text{ecc}(P) = \frac{\lambda_{\max}(P)}{\lambda_{\min}(P)} \leq k \quad \Leftrightarrow \quad \lambda_{\max}(P) - k \cdot \lambda_{\min}(P) \leq 0$$

where k is the positive upper bound. The left side of the last constraint defines a convex function (convex – concave = convex). In Figures 2a-2c, quadratic classification is applied to the same artificial data used in Figure 1. In this case, the quality of linear and quadratic classification is almost identical; the number of misclassified points is 7 in the linear case and 5-6 in the quadratic case. It is clear that (LSV) is a special case of (QC). It is certainly also possible to require the other set of points (+) to lie inside the ellipsoid (Figure 3).

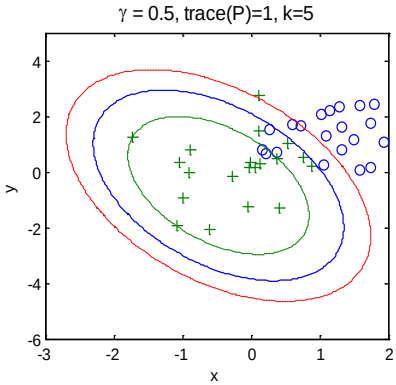

Figure 3: Illustration of quadratic classification around the other group.

Sparse principal component analysis

Principal component analysis is a well-established tool for analysing high dimensional data by reducing it to a lower dimension. Let A be any mean centered m/n matrix encoding m samples of n variables. The principal components are linear combinations of the original variables that point in orthogonal directions explaining as much of the variance in the data matrix A as possible. The weights of the original variables in the principal components are the loadings. Numerically, a full PCA involves a singular value decomposition of the data

Reference paper as:
Pörn, R. and Lillhonga, T. (2012). Applications of semidefinite programming in chemometrics, in:
Proceedings of the 15th International Conference on Near Infrared Spectroscopy,
Edited by M. Manley, C.M. McGoverin, D.B. Thomas and G. Downey, Cape Town, South Africa, pp. 81-86.

matrix. The components are linear combinations of all original variables, *i.e.* most of the loadings are non-zero. PCA facilitates model interpretation, visualisation and analysis by condensing the information to only a few components but the components themselves are still constructed using all original variables and may sometimes be difficult to interpret. In many applications, the coordinate axes involved in the components have a direct physical interpretation. In finance they can be specific assets, in biological applications specific genes and in spectroscopy individual wavelengths. In problems like these it may be desirable to seek a trade-off between the two conflicting goals, namely *statistical fidelity* (explaining most variance) and *interpretability* (simple structure in components).

Sparse PCA has been an active research topic during the last decade. The first ad hoc approach based on simple thresholding was proposed in 1995 by Jolliffe and Cadima[5]. The SCoTLASS algorithm by Jolliffe minimises the Rayleight quotient of the covariance matrix along with a Lasso penalty[6]. Shen and Huang use the SVD to compute low rank approximations of the data matrix with different penalties[7]. The two papers by d'Aspremont *et al.* develop a semidefinite approach to sparse PCA[1,8]. We review some results here.

Given a vector $x \in R^n$. Let card(x) denote the cardinality of the vector x, that is the number of non-zero elements in x. Consider the problem of computing a cardinality constrained eigenvector that corresponds to the maximal eigenvalue of the covariance matrix

(6a-b)
$$\begin{aligned} \max \quad & x^T \Sigma x \\ & x^T x = 1 \\ & \text{card}(x) \le k \end{aligned} \qquad\qquad \begin{aligned} \max \quad & \text{trace}(\Sigma X) \\ & \text{trace}(X) = 1 \\ & \text{card}(X) \le k^2 \\ & X = xx^T \end{aligned}$$

If k=n in (6a) we obtain (1). This is a hard combinatorial problem with exponential complexity. This means that large instances are impossible to solve in reasonable time. The basic idea is now to construct a semidefinite relaxation of this problem following the basic steps (1)-(3). In this case it will be a pure relaxation that only approximates the result in (6). A rank-1 constrained problem that is equivalent to (6a) is given in (6b). The objective and the first constraint are now linear and the rank-1 constraint can be relaxed. What about the cardinality constraint? The authors[1,8] use a standard strategy to obtain a sparse vector, they replace the non-convex cardinality constraint by the weaker but convex constraint; the sum of all absolute values of elements of X should be less than or equal to the target cardinality k^2. A semidefinite relaxation of problem (6) is now defined as

(8a-b)
$$\begin{aligned} \max \quad & \text{trace}(\Sigma X) \\ & \text{trace}(X) = 1 \\ & e^T |X| e \le k^2 \\ & X \succeq 0 \end{aligned} \qquad\qquad \begin{aligned} \max \quad & \text{trace}(\Sigma X) - \rho \cdot e^T |X| e \\ & \text{trace}(X) = 1 \\ & X \succeq 0 \end{aligned}$$

The problem to the left has a target sparsity parameter k and a sparsity constraint. The problem to the right penalises sparsity in the objective using a positive parameter ρ to control the magnitude of the penalty. For a certain pre-specified target sparsity k, it is possible to find a ρ-value that matches that sparsity. The SDP approach can be used to obtain sparse components for small and medium sized problems. Orthogonality between components can also be enforced in this setting.

A related solution approach for formulation (8b) is developed in Journee *et al.*[9]. The method is called GPower to resemble its close connection to the traditional power method for computing eigenvalues. This method is very fast and can be applied to large scale problems. The method works in three steps: 1) initialisation; 2) fast optimisation to obtain a good sparsity pattern (position of zero elements) and 3) ordinary SVD on the reduced matrix (zero columns removed). The optimisation problem in step 2 has a close connection to (8b). The optimisation has a time complexity of $O(mn)$ per component. We use this approach for large scale imaging data in next section.

Reference paper as:
Pörn, R. and Lillhonga, T. (2012). Applications of semidefinite programming in chemometrics, in:
Proceedings of the 15th International Conference on Near Infrared Spectroscopy,
Edited by M. Manley, C.M. McGoverin, D.B. Thomas and G. Downey, Cape Town, South Africa, pp. 81-86.

The GPower method applied to imaging data

The test data consists of hyperspectral imaging data of sandpaper with three different backing materials. The objective is to compare sparse PCA with full PCA. The sisuCHEMA hyperspectral camera (Specim, Spectral Imaging Ltd, Oulu, Finland) with a wavelength region of 1000 nm – 2498 nm (239 wavelengths) was used.

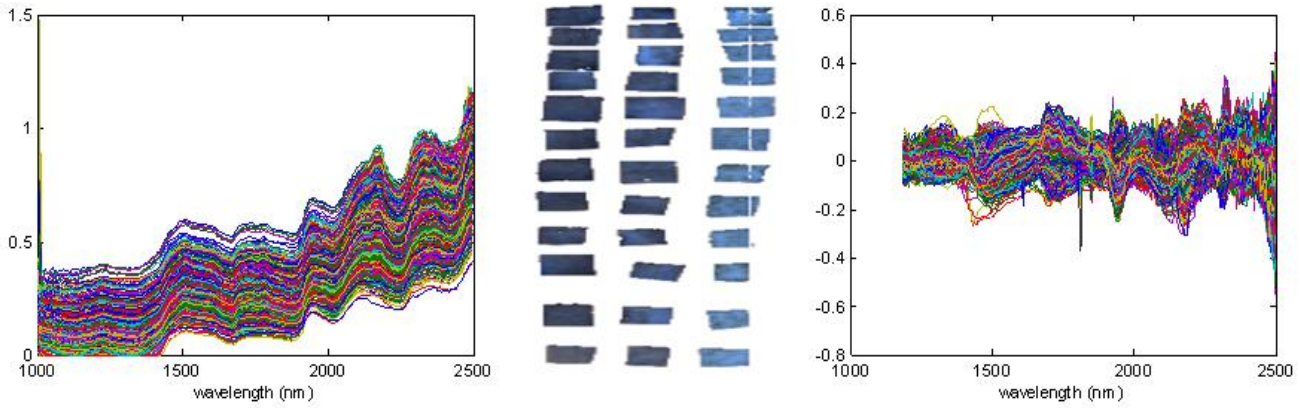

Figure 4. 12x3=36 samples of glued sandpaper with three different backing materials and curing times from 1-8 hours. Score plots from Evince (middle); Original NIRS (12801 pixels, background removed) (left); Spectra pre-processed with snv and mean centered (210 wavelengths and 12801 pixels) (right).

The experiment shows how to control the sparsity of the loading vector and how the explained variance depends on sparsity. We extract the first principal component for different values of the penalty parameter and register the corresponding sparsity value in percent; 0% means no sparsity (a dense loading vector) and 100% mean maximal sparsity (0-1 non-zeros). The results are given in Figure 5. For example, a parameter value of 0.2 gives a sparsity of about 40%. This relationship is naturally data dependent. We also illustrate how the explained variance decreases along with increased sparsity. With a sparsity of 0-40%, the first sparse principal component explains as much as a dense component; for higher sparsity values, the degree of explanation starts to decrease.

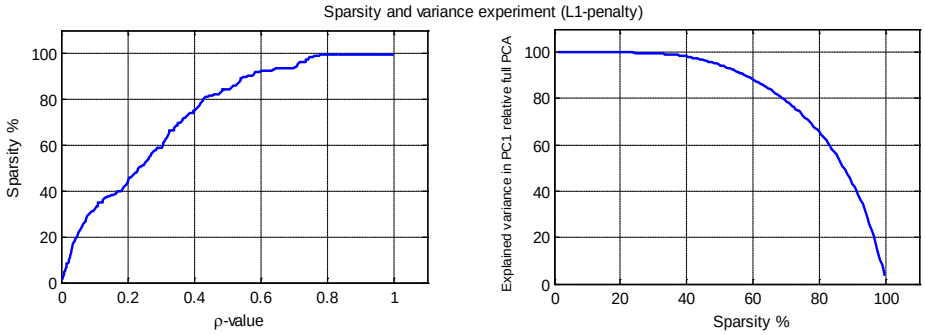

Figure 5. Sparsity % for PC1 versus ρ -value (left). Variance explained in PC1 in % relative to PC1 in full PCA (right).

In the next experiment we did sparse PCA on the imaging data and summarised the results in Figure 6. Three sparse components were extracted with sparsity of about 60% (~90 wavelengths). Three different backing materials were used and this is clearly reflected in the two distinct groups in the first score plot. Two of the backing materials were very similar. The score plot is essentially identical to that of the full PCA. The explained variance is shown in the scree plot. For sparse PCA, the measurement of adjusted variance is used according to the recommendation in Shen and Huang[7]. This is due to the possibility of non-orthogonal sparse components. The scores in PC1 are very similar in both the sparse and full cases. The sparse loadings contain a few major peaks that are formed by adjacent wavelengths. In this case, the sparse loadings have similar structure to that of the dense loadings.

Results and Discussion

In this paper we studied semidefinite programming as a theoretical framework for applications in chemometrics. Two examples were given: quadratic classification and sparse principal components. Quadratic classification can be modelled and solved efficiently using SDP software. Sparse PCA results in a

Reference paper as:
Pörn, R. and Lillhonga, T. (2012). Applications of semidefinite programming in chemometrics, in:
Proceedings of the 15th International Conference on Near Infrared Spectroscopy,
Edited by M. Manley, C.M. McGoverin, D.B. Thomas and G. Downey, Cape Town, South Africa, pp. 81-86.

SDP problem that can be solved efficiently up to medium sized instances. For large scale examples, we turn to the recent GPower method that solves an optimisation problem closely related to the SDP-relaxation. The method is fast and reliable on a large-scale data set from hyperspectral imaging. The technique is efficient from a data reduction and interpretability perspective. Calibration and regression have not been investigated further in this study. All computations very carried out in MATLAB v. 2.10; 7.5.0 2007b (The MathWorks, Massachusetts, USA). The SDPs where solved by the CVX toolbox (cvxr.com/cvx) and sparse principal components were obtained by the GPower method (www.montefiore.ulg.ac.be/~journee).

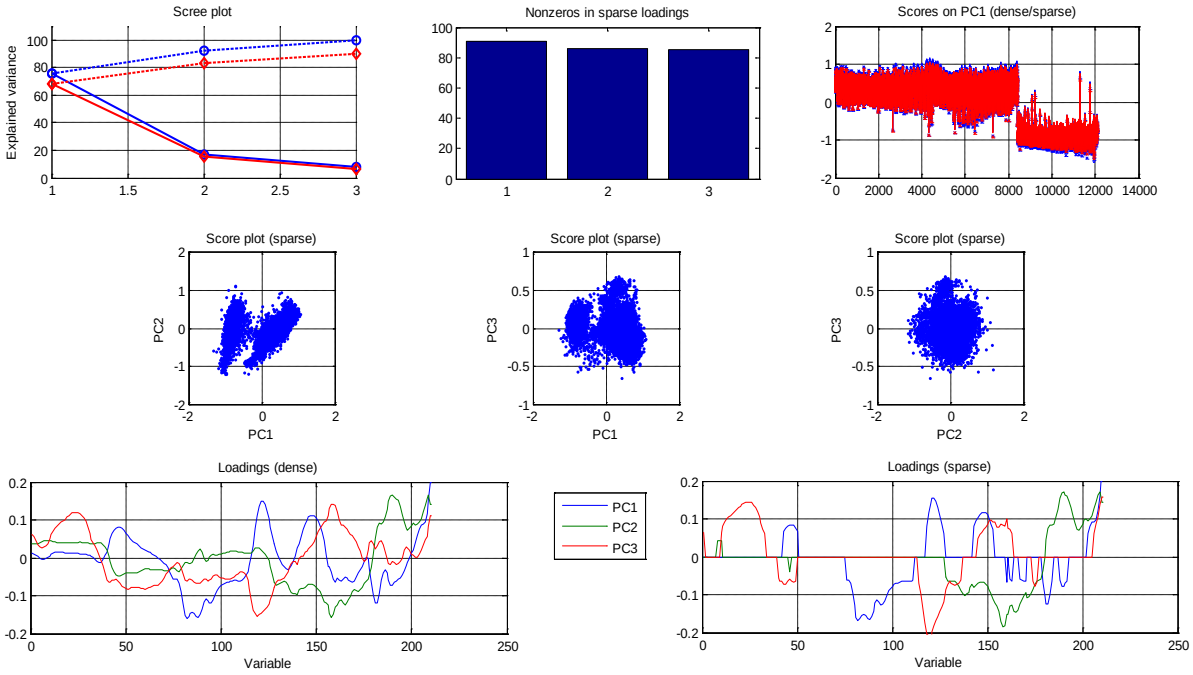

Figure 6. Scree plot: variance and cumulative variance for components 1-3 (sparse PC with red lines, full PCA with blue lines). Scores on PC1: Sparse PC in red and dense PC in blue. Score plots for sparse PC 1-3. Loading plots for dense PC (left) and sparse PC (right). Variables 0-210 correspond to wavelengths 1180-2498 nm.

Sparse principal components are an interesting and useful application. Sparse components may enhance interpretation and lead to significant data compression, especially in large scale hyperspectral imaging applications. Data reduction is important both from a speed and a storage point of view. It is possible to replicate the performance of full PCA classification by using only 50% of the wavelengths. The extraction of sparse principal components may also be used as a pre-processing step for principal component regression (PCR) or partial least squares (PLS) regression.

Acknowledgements

The authors are grateful for financial support from the Botnia Atlantica project Field-NIRce.

References

1. S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, Cambridge, UK (2004).
2. A. d'Aspremont, L. El Ghaoui, M.I. Jordan and G.R.G. Lanckriet, *SIAM Rev.* **49**, 434-448 (2007).
3. K. Weinberger and L. Saul, *J. Mach. Learn. Res.* **10**, 207-244 (2009).
4. J. Luts, F. Ojeda, R. Van de Plas, B. De Moor, S. Van Huffel and J. Suykens, *Anal. Chim. Acta* **665**, 129-145 (2010).
5. J. Cadima and I.T. Jolliffe. *J. Appl. Stat.* **22**, 203-214 (1995).
6. I.T. Jolliffe, N.T. Trendafilov and M. Uddin, *J. Comput. Graph. Stat.* **12**, 531-547 (2003).
7. H. Shen and J.Z. Huang, *J. Multivariate Anal.* **99**, 1015-1034 (2008).
8. A. d'Aspremont, F.R. Bach and L. El Ghaoui, *J. Mach. Learn. Res.* **19**, 1269-1294 (2008).
9. M. Journée, Y. Nesterov, P. Richtárik and R. Sepulchre, *J. Mach. Learn. Res.* **11**, 517-553 (2010).

Reference paper as:
Pörn, R. and Lillhonga, T. (2012). Applications of semidefinite programming in chemometrics, in:
Proceedings of the 15th International Conference on Near Infrared Spectroscopy,
Edited by M. Manley, C.M. McGoverin, D.B. Thomas and G. Downey, Cape Town, South Africa, pp. 81-86.