sBarse: Sparse biplots component analysis

Doyo Gragn and Nickolay T. Trendafilov

Department of Mathematics and Statistics,

The Open University, Walton Hall, MK7 6AA, UK

November 9, 2009

Abstract

A method for facilitating the principal components interpretation is proposed. The principal components loadings are simplified (sparsified) by exploring the approximation features of a list of sparse biplots. The resulting sparse (sBarse) loadings matrix have maximum product of adjusted variance and RV-coefficient. By construction, each original variable corresponds to only one sBarse component, which makes the solution transparent and uniquely interpretable. In addition, the method finds the appropriate number of components to retain. A well-studied data set is used for numerical illustration. Then the proposed method is applied to a real gene expression data set with $p \gg n$. Each sBarse component contains a cluster of genes, which are non-overlapping with the genes in the other components.

Keywords: Sparse principal components; Biplots; RV-coefficient; Gene expression data; $p \gg n$.

1 Introduction

There exists a number of methods for reducing the data dimensionality (Seber, 2004). The most popular and efficient method is principal component analysis (PCA) (Jolliffe, 2002). It performs a linear mapping of the data to a low dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized.

The principal components (PCs) are really useful if they can be easily interpreted. Unfortunately, each PC is a weighted sum of all original variables which makes the interpretation difficult, ambiguous, and/or even impossible. Traditionally, PCs are considered easily interpretable if there are plenty of small component loadings indicating the negligible importance of the corresponding original variables (Jolliffe, 2002). Thus, the PCs simplicity and interpretability is associated with their sparseness. The "classical" way of approximating PCs for simple interpretation is to ignore loadings whose absolute values are below some specified threshold. However, Cadima and Jolliffe (1995) argue that ignoring small-magnitude loading in the interpretation of PCs is misleading, especially for PCs computed from covariance matrix.

Many approaches are already proposed to aid the PCs interpretation. The most influential ones are briefly reviewed in the sequel. Historically, the first approach is the simple structure rotation (Jolliffe, 2002). This is simply a change of the coordinate axes according to certain simplicity criterion. The main drawback of this approach is that the rotated loadings are usually still difficult to interpret. Its application is additionally complicated by the huge number of simplicity criteria to choose from. Another important shortcoming is that the rotated components lack the nice PCs properties to be uncorrelated and explain successively decreasing amount of variance.

Thus, it seems a better alternative to develop a modified PCA that explicitly produces simple PCs. The first method to directly construct simplified (sparse) components (SCs) was proposed by Hausman (1982) which finds PC loadings from some prescribed subset of values, say $S = \{-1, 0, 1\}$. Later this idea was extended to arbitrary integers (Vines, 2000). Cadima and Jolliffe (1995) were the first to attempt modifying the original PCs to satisfy certain simplicity criterion (varimax) still explaining successively decreasing portion of the variance. The same line of achieving interpretability was further pursued in SCoTLASS where the loadings satisfy the additional LASSO constraint and thus, many of them are driven to exact zeros (Jolliffe *et al.*, 2003). The sparse PCA (SPCA) proposed by Zou *et al.* (2006) further elaborates the LASSO idea leading to a very efficient numerical procedure. Witten *et al.* (2009) recently propose a new algorithm for solving the SCoTLASS problem. However, the same conceptual difficulty remains: the right choice of the LASSO threshold that compromise between sparseness and explained variance. The LASSO constraint is replaced by cardinality constraint (number of zero loadings per component) and the problem is transformed into a semidefinite programming by d'Aspremont *et al.* (2007). Despite the advanced numerical technique, the choice of the cardinality presents very much the same problem as with the LASSO threshold in SCoTLASS and SPCA. Probably the most elegant approach that swiftly treats both LASSO and cardinality constraints was recently proposed by Journée *et al.* (2008). Nevertheless, the LASSO/cardinality related approaches to sparseness are numerically demanding while leaving freedom for subjective interpretation. Necessarily, they are followed by some kind of validation of the threshold/cardinality, which may not be feasible for large data.

A very simple way to avoid the LASSO related problems was achieved by Chipman and Gu (2005). They introduce three types of "interpretable" components. Then, the best SC to be chosen is the "interpretable" one that has the least angle with the original PC. This idea is further studied and elaborated by Anaya-Izquierdo *et al.* (2008) without providing efficient numerical implementation.

Rousson and Gasser (2004) propose to find block SCs by classifying first the original variables into disjoint groups approach, and then finding components from regressing the original variables on the preceding sparse components.

Another promising approach for obtaining SCs is based on the spectral bounds of submatrices of the sample correlation matrix. The idea is to identify the subset of m variables explaining the maximum variance among all possible subsets of size m and replace the loadings of the rest $p - m$ variables by 0s. It is first applied in quite latent form by Cadima and Jolliffe (2001) for variable selection. The idea is really developed and further used to obtain a greedy sparse PCA algorithm by Moghaddam *et al.* (006b). A valuable comparative study of this approach and the semidefinite programming is provided in d'Aspremont *et al.* (2008). From data analysis point of view, the main difficulty is again the choice of m that compromises with the portion of the explained variance. Farcomeni (2009) suggested maximizing the explained variance penalized by the cardinality of the SCs sought.

The approaches for sparse and interpretable PCA can be readily extended to linear discriminant analysis: the spectral bounds approach – in Moghaddam *et al.* (006a), the LASSO approach – in Trendafilov and Jolliffe (2007), the PCs least angle approximation – in Trendafilov and Vines (2009), and the block components idea – in Sabatier and Reynès (2008). Recently, the LASSO approach is also extended to canonical correlation analysis by Lykou and Whittaker (2009) and Witten *et al.* (2009).

In this paper a very simple method for sparse reduction of dimensionality is proposed. It is called sBarse for short. Sparse loadings are constructed from the biplots of the input data: either the data matrix or the sample correlation matrix. The resulting sBarse components have orthogonal loadings, i.e. each original variable corresponds to only one sBarse component, and thus, leading to easily interpretable components. This is in contrast with many existing methods producing non-orthogonal SCs, e.g. Chipman and Gu (2005); d'Aspremont *et al.* (2007); Moghaddam *et al.* (006b); Witten *et al.* (2009) etc. The sparseness of the sBarse solution and the number m of the SCs involved are chosen to maximize the adjusted variance of the sBarse components and be as close as possible to the input data in terms of the RV-coefficient (Robert and Escoufier, 1976). Thus the sBarse components found do *not* rely on any tuning parameters.

The paper is organized as follows. An intuitive introduction to the sBarse method is given in Section 2, followed by a more formal treatment in Section 3. In Section 4, the sBarse method is first applied and compared with other similar methods on the benchmark Jeffers's Pitprop data (Jeffers, 1967). It is demonstrated that the sBarse solution is superior in many ways to the existing methods. Then, the sBarse method is applied to study a real gene expression data set from breast cancer cell, a case where the number of variables is by far larger than the samples. This data set is used by Chin *et al.* (2006) and is freely available from http://icbp.lbl.gov/breastcancer/. For comparison, we use the subset of the gene expression data set considered in Witten *et al.* (2009). The sBarse solution is compared to the one obtained by Witten *et al.* (2009) . More discussion and conclusions are given in Section 5.

2 Simplified principal components

2.1 Rationale

There are a number of different ways to achieve PCs simplification as listed in Section 1. The proposed new method produces simplified loadings for all components simultaneously in contrast to most of the existing methods where each PC is simplified separately from the others.

Let $\mathbf{A}$ be a $p \times p$ matrix of PC loadings, whose jth column represents the jth eigenvector of the correlation matrix $\mathbf{R}$ corresponding to the jth largest eigenvalue λ_j, $j = 1, 2, \ldots, p$. Then each loading a_{ij} in the matrix $\mathbf{A}$ represents the contribution of the ith original variable x_i in the jth PC. The aim is to simplify the loadings by comparing the contribution of a variable to each of the PCs, so that the resulting SCs are easier to interpret. The idea is that a variable will be retained only in this SC in which it is most important. This can be easily achieved as follows.

Assume that each row of $\mathbf{A}$ represents a point on a p-dimensional Euclidean space. The proposed simplification mechanism is to approximate the ith row-vector of $\mathbf{A}$ by the nearest p-dimensional unit-vector. This requires finding the least Euclidean distance between the ith row-vector of $\mathbf{A}$ and all possible unit-vectors.

There are $2p$ such unit-vectors available, $\mathbf{e}_k$ and $-\mathbf{e}_k$ for k=1, 2,. . . , p. For instance, in a 2-dimensional space, a row-vector in a 2×2 matrix of loadings $\mathbf{A}$ can be approximated by either of the following: $(1, 0)$, $(0, 1)$, $(-1, 0)$, or $(0, -1)$. That is, after approximation, only one of the coefficients on the row of $\mathbf{A}$ take the value 1 (or -1 if the original loading is negative) and the remaining coefficients take 0. Note that $(0, 0)$ is not a possible option, as we assume that each variable has a contribution in explaining a PC.

From the property of PCs, the jth PC has the jth largest variance, which is given by the eigenvalue. However, the above simplification procedure uses only the unweighted loadings (or weights equal to 1) and does not take into account the variances of the PCs. This does not preserve the property of PCs that the first few PCs explain the majority of the variation in the data. Hence, this weighting property can be incorporated into the method by multiplying the loadings of the PCs by the square root of the corresponding eigenvalues. Since the eigenvalues are in decreasing order of magnitude, more weights are being given to the first few PCs.

Consider the ith row-vector of $\mathbf{A}$ and let $b_{ij} = \sqrt{\lambda_j} \times a_{ij}$ be the ith element in the jth column of weighted matrix $\mathbf{B}$. The row-vector of weighted loadings becomes

$$\mathbf{b}_i^\top = [\sqrt{\lambda_1} a_{i1} \quad \sqrt{\lambda_2} a_{i2} \quad \ldots \quad \sqrt{\lambda_p} a_{ip}]. \tag{1}$$

Let δ_{kj} be the Kronecker delta for $j = 1, 2, \ldots, p$ and $k = 1, 2, \ldots, p$. The Euclidean distances, between the ith row-vector (1) and each of the $2p$ unit-vectors $\mathbf{e}_k$ and $-\mathbf{e}_k$ are:

$$\sum_j^p (b_{ij} \pm e_{kj})^2, k = 1, 2, \ldots, p \tag{2}$$

with $e_{kj} = \delta_{kj}$

The unit-vector $\mathbf{e}_k$ or $-\mathbf{e}_k$ that gives the smallest value of (2) is considered to approximate the row-vector $\mathbf{b}_i^\top$. The same is repeated for all rows $i = 1, \ldots, p$. Each column of the resulting simplified matrix is then normalized by dividing each element in the column by the square-root of the number of non-zero elements in the column. This gives a matrix of loadings for the simplified components.

For the ith row of $\mathbf{B}$, comparing the distances in (2) is equivalent to comparing $|b_{ij}|$ (or b_{ij}^2) for $j = 1, \ldots, p$. It turns out that, the largest (in absolute value) of the weighted loadings in the ith row takes a value of 1 (or -1 if the original loading is negative) and 0 otherwise. Then, each optimal unit-vector is inserted in a resulting matrix $\mathbf{Y}$ row after row. Finally, the columns of $\mathbf{Y}$ are normalized and called sBarse components.

The aim of the sBarse procedure is to make the interpretation of PCs easier. However, there is one more advantage gained by the method: it can help to find the appropriate number of components to retain. Indeed, rewrite (1) as $[\lambda_1 a_{i1}^2 \quad \lambda_2 a_{i2}^2 \quad \ldots \quad \lambda_p a_{ip}^2]$. The elements of this vector are the values to be compared for deciding on the approximating unit-vector. Since the eigenvalues are in decreasing order of magnitude, let $\lambda_2 = \theta_1 \lambda_1, \lambda_3 = \theta_2 \lambda_2, \ldots, \lambda_p = \theta_{p-1} \lambda_{p-1}$ where $0 < \theta_i < 1$ for $i = 1, 2, \ldots, p-1$. Substitution indicates that the values to be compared are the elements of the vector

$$[a_{i1}^2 \quad \theta_1 a_{i2}^2 \quad \theta_1 \theta_2 a_{i3}^2 \quad \ldots \quad (\theta_1 \theta_2 \cdots \theta_{p-1}) a_{ip}^2].$$

The increasing number of θ_i's multiplying the coefficients leads to a very small product in the last few columns. Thus, the sBarse procedure reduces the last $p - m$ squared loadings (and hence the last $p-m$ columns in all rows) to 0 and so the matrix of original loadings can be approximated only by the first m orthogonal vectors of SCs. That means, the points of the data matrix can be represented in a reduced m-dimensional subspace of the p-dimensional space. This estimation of m might be used as an alternative to the scree plot and the percentage of the explained variation for deciding the number of PCs to retain (Jolliffe, 2002, p.115).

2.2 Correlation as a criterion

The sBarse procedure outlined in section 2.1 can be described in the following more meaningful way. Let $R^2_{i:p}$ denotes the squared multiple correlation between the ith standardized variable and the PCs, $y_1, y_2, \ldots, y_p$. Since the PCs are uncorrelated, the squared correlation can be written as r^2_{ij}, for $j = 1, \ldots, p$, where r^2_{ij} is the squared linear correlation between x_i and y_j. Recall that the pair (λ_j, a_{ij}) represents the variance and the ith loading for the jth PC. Then, $r^2_{ij} = \lambda_j a^2_{ij}$ (Jolliffe, 2002, p.25) which is the same as the squared weighted loading, b^2_{ij}, considered in section 2.1.

Hence, for the ith row of $\mathbf{B}$, the way of describing its sparsefication is to associate variable x_i to the jth PC y_j with which the variable has the largest squared correlation, r^2_{ij}; i.e., a variable is being related to a specific PC based on its explanatory power with respect to the PC.

The following example illustrates the sBarse procedure using a well known data set.

Example 1: The Pitprop data contains 13 variables measured for 180 pitprops cut from Corsican pine timber (Jeffers, 1967). This data set is already a standard example to consider in any work on sparse approximation of PCA.

The loadings of the first six PCs computed from the correlation matrix of the Pitprop data and the corresponding variance explained by them are given in the first six columns of Table 1. Then, the sBarse method is applied and the solution is given in the last six columns of Table 1. The last seven SCs produced (not shown) are identically zero. Thus, the interpretation of the Pitprop data will be based on the first six sBarse components – the same number of components chosen by Jeffers (1967) for further analysis.

Table 1: *Loadings of the first six PCs and the corresponding SCs, Jeffers's Pitprop data*

	Principal Components						Sparse Components					
Variable	1	2	3	4	5	6	1	2	3	4	5	6
1	-.40	.22	-.21	-.09	-.08	.12	-.41	0	0	0	0	0
2	-.41	.19	-.24	-.10	-.11	.16	-.41	0	0	0	0	0
3	-.12	.54	.14	.08	.35	-.28	0	.71	0	0	0	0
4	-.17	.46	.35	.05	.36	-.05	0	.71	0	0	0	0
5	-.06	-.17	.48	.05	-.18	.63	0	0	.71	0	0	0
6	-.28	-.01	.48	-.06	-.32	.05	0	0	.71	0	0	0
7	-.40	.19	.25	-.07	-.22	.00	-.41	0	0	0	0	0
8	-.29	-.19	-.24	.29	.19	-.06	-.41	0	0	0	0	0
9	-.36	.02	-.21	.10	-.10	.03	-.41	0	0	0	0	0
10	-.38	-.25	-.12	-.21	.16	-.17	-.41	0	0	0	0	0
11	.01	.21	.07	.80	-.34	.18	0	0	0	1	0	0
12	.12	.34	.09	-.30	-.60	-.17	0	0	0	0	-1	0
13	.11	.31	-.33	-.30	-.08	.63	0	0	0	0	0	1
%var	32.4	18.3	14.4	8.5	7.0	6.3	29.1	14.6	10.6	7.7	7.7	7.7
%Cvar	32.4	50.7	65.1	73.6	80.6	86.9	29.1	43.7	54.3	62.0	69.7	77.4

3 Computing sBarse components

3.1 Biplots and their goodness-of-fit

The weighing scheme used in the sBarse method is closely related to the biplots and the measures of their goodness-of-fit to the data (Gabriel, 1971; Gower and Hand, 1996).

Let $\mathbf{X}$ be a standardized $n \times p$ data matrix of rank r. Then the singular value decomposition (SVD) of $\mathbf{X}$ is given by

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{A}^\top, \tag{3}$$

where $\mathbf{U}$ and $\mathbf{A}$ are $n \times r$ and $p \times r$ orthonormal matrices, and $\boldsymbol{\Sigma}$ is the $r \times r$ diagonal matrix of singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$. Let $\boldsymbol{\Sigma}^\beta$ $(0 \leq \beta \leq 1)$ be the diagonal matrix whose elements are $\sigma_1^\beta, \sigma_2^\beta, \ldots, \sigma_r^\beta$ and (3) be rewritten as:

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}^{1-\beta}\boldsymbol{\Sigma}^\beta\mathbf{A}^\top. \tag{4}$$

Let $\mathbf{U}_m = [\mathbf{u}_1, ..., \mathbf{u}_m]$, $\mathbf{A}_m = [\mathbf{a}_1, ..., \mathbf{a}_m]$ and $\boldsymbol{\Sigma}_m = \begin{pmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_m \end{pmatrix}$ for any $m \in [1, r]$. Denote $\mathbf{G}_{\beta,m} = \mathbf{U}_m\boldsymbol{\Sigma}_m^{1-\beta}$ and $\mathbf{H}_{\beta,m} = \mathbf{A}_m\boldsymbol{\Sigma}_m^\beta$. Then, the following approximation holds:

$$\mathbf{X} = \mathbf{G}\mathbf{H}^\top \approx \mathbf{G}_{\beta,m}\mathbf{H}_{\beta,m}^\top. \tag{5}$$

The matrices $\mathbf{G}_{\beta,m}$ and $\mathbf{H}_{\beta,m}$ are called biplot factors and their rows, called biplots, are the markers for the n rows (observations) and p columns (variables) of $\mathbf{X}$. Biplots are used to approximate the data $\mathbf{X}$ and can be constructed with any factors $\mathbf{G}_{\beta,m}$ and $\mathbf{H}_{\beta,m}$, and with any choices of $\beta \in [0,1]$ and $m \leq r$. Interpretation of the most important biplots with $\beta = 0, \frac{1}{2}$ and 1 is given in Jolliffe (2002).

Biplots are also used to approximate the sample correlation matrix $\mathbf{R} = \mathbf{X}^\top \mathbf{X}$ (Gabriel, 1971; Gower and Hand, 1996, Ch 2, 11). They are called correlation biplots and can also be constructed with any choice of $\beta \in [0,1]$ as above, but with a single biplot factor $\mathbf{H}_{\beta,m} = \mathbf{A}_m \mathbf{\Sigma}_m^\beta$. For example, the choice of $m = 2$ and $\beta = 1$ gives the biplot factor $\mathbf{H}_2 = \mathbf{A}_2 \mathbf{\Sigma}_2$, which is the best two-dimensional least squares approximation of $\mathbf{R} \approx \mathbf{H}_2 \mathbf{H}_2^\top$. This follows from the eigenvalue decomposition (EVD) of the sample correlation matrix $\mathbf{R} = \mathbf{A}\mathbf{\Sigma}^2\mathbf{A}^\top = \mathbf{H}\mathbf{H}^\top$, where $\mathbf{\Lambda} = \mathbf{\Sigma}^2$ contains the eigenvalues of $\mathbf{R}$. In general, consider the following biplot factor $\mathbf{B}_{\alpha,m} = \mathbf{A}_m \mathbf{\Lambda}_m^\alpha$ of rank m with $\alpha \in [0,1]$. Then, the biplot approximation of the correlation matrix $\mathbf{R}$ is given by

$$\mathbf{R}_{\alpha,m} = \mathbf{B}_{\alpha,m}\mathbf{B}_{\alpha,m}^\top = \mathbf{A}_m \mathbf{\Lambda}_m^\alpha \mathbf{\Lambda}_m^\alpha \mathbf{A}_m^\top = \mathbf{A}_m \mathbf{\Sigma}_m^{2\beta} \mathbf{\Sigma}_m^{2\beta} \mathbf{A}_m^\top \ , \tag{6}$$

where the choice $\alpha = \frac{1}{2}$ gives the best least-squares approximation to $\mathbf{R}$ of rank m. The standard biplots aim low-dimensional data visualization, but the aim of the sBarse method is primarily sparse and cheap loadings matrix. For this reason a wider interval for the power α is adopted. However, increasing the upper limit for α beyond 1 is not reasonable as this will result in one or very few PCs with poor approximation properties. As with the standard biplots one can use a range of values α. Considering several $\alpha \in [0,1]$ and excluding the improper solutions (containing zero column(s) followed by non-zero columns) gives a list of admissible solutions such that the user can choose the most satisfying one. It is natural to base this choice on the amount of the explained variance by the sBarse components and/or on their approximation power measured by Gabriel (2002) as the goodness-of-fit of the biplot approximation $\mathbf{R}_{\alpha,m}$ to $\mathbf{R}$ and making use of the RV-coefficient (Robert and Escoufier, 1976):

$$\mathrm{RV}^2(\mathbf{R}, \mathbf{R}_{\alpha,m}) = \frac{\mathrm{trace}^2(\mathbf{R}\mathbf{R}_{\alpha,m})}{\mathrm{trace}(\mathbf{R}^2)\mathrm{trace}(\mathbf{R}_{\alpha,m}^2)} . \tag{7}$$

The RV values lie in the interval $[0,1]$. Values close to 1 mean better approximation.

3.2 Sparse biplots and sBarse components

The weighing scheme of the sBarse method outlined in Section 2.1 uses only $\alpha = 0.5$. As with the biplots one can use a range of values α.

In general, the small values of α lead to solutions with more sBarse components, while the bigger α's correspond to fewer sBarse components, which naturally follows from Section 2.1. An interesting question is: how many α's to take to be sure that no valuable solution is missed? The answer is: not too many, because large intervals of α's correspond to a single set of sBarse components.

Example 1 (continued): For the Pitprop data, the algorithm uses $\alpha \in [0, 1]$ with a step of .02. It is found that for $\alpha = 0.4$ the algorithm produces the best proper solution with six sBarse components (the last six columns of Table 1), accounting for 77% of the total variation. This value of α is not uniquely defined; other values of α, say $\alpha = 0.5$, also results in the same solution. Hence, the best solution corresponds to an interval of α values.

It should be noted that all methods applied to the Pitprop data (d'Aspremont *et al.*, 2007; Farcomeni, 2009; Jolliffe *et al.*, 2003; Moghaddam *et al.*, 006b; Zou *et al.*, 2006) a priori take and work with the first six sparse components explaining a reasonable portion of the original variance, while the proposed method *finds* the appropriate number (again 6) of the sBarse components. More details on the sBarse components solution of the Pitprop data and comparison with other sparse solutions are considered in Section 4.

The sBarse orthogonal loadings are correlated. To take this into account, it is suggested to replace the standard variances by the corresponding 'adjusted' variances (Zou *et al.*, 2006). The essence of the sBarse algorithm is to produce a list of admissible solutions and rank them according to their adjusted variances. It seems also reasonable to take into account the goodness-of-fit of the produced solutions. Thus, the admissible solutions obtained from sBarse algorithm will be ranked according to the product of their adjusted variance and their RV-coefficients (7).

The computational procedure can be summarized as follows. The sBarse method finds a set of admissible sparse matrices $\mathbf{Y}_m$ with elements from $\{-1, 0, 1\}$ and certain rank m, which approximate the biplot factor $\mathbf{B}_m = \mathbf{A}_m \mathbf{\Lambda}_m^\alpha$ for $\alpha \in [0, 1]$, i.e. $\mathbf{B}_m \approx$

$\mathbf{Y}_m$. The biplot factors are sparsefied using the procedures proposed in Section 2. For interpretation purposes the approximation $\mathbf{Y}_m$ of the biplot factor $\mathbf{B}_m$ should come as a product of an orthonormal matrix (of sparse loadings) multiplied by a diagonal matrix of "variances". According to Section 2, $\mathbf{Y}_m$ is first normalized such that $\mathbf{Y}_m^\top \mathbf{Y}_m = \mathbf{I}_m$ and then, is assigned to be the orthonormal term in the sparse biplot factor. The diagonal term in the biplot factor is simply formed by taking the variances of the new SCs (having sparse loadings $\mathbf{Y}_m$), i.e. the main diagonal of $\mathbf{Y}_m^\top \mathbf{R} \mathbf{Y}_m$. As the new SCs are correlated it is reasonable to replace their variances by the corresponding adjusted variances introduced by Zou *et al.* (2006). Let $\mathbf{C}_m$ be the upper-triangular $m \times m$ factor of the Cholesky decomposition of $\mathbf{Y}_m^\top \mathbf{R} \mathbf{Y}_m$, i.e. $\mathbf{Y}_m^\top \mathbf{R} \mathbf{Y}_m = \mathbf{C}_m^\top \mathbf{C}_m$. Then, the diagonal matrix composed by the main diagonal of $\mathbf{C}_m$ and denoted by diag($\mathbf{C}_m$), is taken to be the second term in the sparse biplot factor. The goodness-of-fit of this approximation of $\mathbf{R}$ is given by (Gabriel, 2002):

$$\begin{aligned} \mathrm{RV}^2\left(\mathbf{R}, \mathbf{Y}_m \mathrm{diag}^2(\mathbf{C}_m)\mathbf{Y}_m^\top\right) &= \frac{\mathrm{trace}^2\left(\mathrm{diag}(\mathbf{Y}_m^\top \mathbf{R} \mathbf{Y}_m)\mathrm{diag}^2(\mathbf{C}_m)\right)}{\mathrm{trace}(\mathbf{\Lambda}^2)\mathrm{trace}\left(\mathrm{diag}^4(\mathbf{C}_m)\right)} \\ &= \frac{\mathrm{trace}^2\left(\mathrm{diag}(\mathbf{C}_m^\top \mathbf{C}_m)\mathrm{diag}^2(\mathbf{C}_m)\right)}{\mathrm{trace}(\mathbf{\Lambda}^2)\mathrm{trace}\left(\mathrm{diag}^4(\mathbf{C}_m)\right)} \quad . \end{aligned} \tag{8}$$

This procedure can be summarized in the following algorithm:

```
$\mathbf{A} \leftarrow$ eigenvectors of $\mathbf{R}$;
$\mathbf{D} \leftarrow$ eigenvalues of $\mathbf{R}$;
set discretization step $\Delta$, say $\Delta = .02$;
for $\alpha = 0 : \Delta : 1$; % start of $\alpha$-loop
    $\mathbf{B} = \mathbf{A}\mathbf{D}^\alpha$;
    sparsify $\mathbf{B}$ in $\mathbf{Y}$;
    check that such sparse $\mathbf{Y}$ does not exist yet;
    check that $\mathbf{Y}$ is admissible, i.e. all first $m(\leq p)$ columns of $\mathbf{Y}$ are non-zero;
```

```
        normalize $\mathbf{Y}_m$, i.e. $\mathbf{Y}_m^\top \mathbf{Y}_m = \mathbf{I}_m$;
        $\mathbf{C}_m^\top \mathbf{C}_m = \mathbf{Y}_m^\top \mathbf{R} \mathbf{Y}_m$ % Cholesky decomposition
        adjvar = $\mathrm{diag}^2(\mathbf{C})$;
        max = adjvar$\sqrt{\texttt{RV}}$;  % RV = RV-coefficient
        if max is the largest for this discretization;
            $\alpha_{\texttt{max}} \leftarrow \alpha$;
        endif
    end % end of $\alpha$-loop
    print max, $\mathbf{Y}_m$ obtained from sparsifying $\mathbf{B} = \mathbf{A}\mathbf{D}^{\alpha_{\texttt{max}}}$;
```

If the raw data matrix $\mathbf{X}$ is available, there is no need to form the sample correlation matrix $\mathbf{R}$ in the above procedure/algorithm. Indeed, let $\mathbf{X}\mathbf{Y}_m = \mathbf{Q}\mathbf{T}$ be the QR decomposition of the new SCs, where $\mathbf{Q}$ is an $n \times m$ orthonormal matrix, and $\mathbf{T}$ – $m \times m$ upper-triangular. By definition (Zou *et al.*, 2006), the adjusted variances of the new SCs are given by $\mathrm{diag}^2(\mathbf{T})$. Thus, the required sparse biplot is $\mathbf{Y}_m \mathrm{diag}(\mathbf{T})$, and the RV-coefficient (8) is found by simply making $\mathbf{C}_m \equiv \mathbf{T}_m$.

4 Application

In this section, some more details are given for the sBarse solutions of the Pitprop data (Jeffers, 1967) considered in the previous sections. Next, a real gene expression data set (Chin *et al.*, 2006) is considered. The sBarse solutions are compared with sparse solutions obtained by other methods.

Example 1 (The Pitprop data, continued): For the Pitprop data, there are 29 admissible solutions (out of 51) obtained by the sBarse algorithm with $\alpha \in [0, 1]$. For many values of α identical sBarse components are found. The sBarse algorithm checks

Table 2: *Admissible sBarse components for the Pitprop data for* $\alpha \in [0, 1]$ and step .02.

Sol	α	RV	Var	Adj	RV×Adj	# sBarse comp.
1	.36	.8580	.7684	.7325	.6285	6
2	.68	.8233	.5938	.5910	.4866	4
3	.92	.7424	.5590	.5497	.4081	4
4	.94	.5829	.4801	.4426	.2580	4
5	.96	.5339	.4428	.4294	.2293	4
6	1.00	.6109	.5016	.4857	.2967	4

and omits them, i.e. the recalculation of their variances, adjusted variances and RV-coefficients are not needed. Thus, for the Pitprop data, there are only 6 admissible different sBarse components, whose characteristics are reported in Table 2. According to the value of the product of total adjusted variance explained times the RV coefficient the best solution has 6 sBarse components. Its loadings were already given in Table 1.

This example shows that the sBarse method is not able to produce sparse loadings for any $m = 1, 2, ..., p$. This is a disadvantage of the method as it might be necessary to have sparse solution with particular number of components, which the method is unable to produce. In the same time, this can be viewed as an advantage of the method as it reduces the freedom of the choice of proper number of components to retain for analysis.

In Table 3 are given the loadings of the first three sBarse components and the corresponding cumulative variances (CV) and adjusted variances (CAV) found by the several different methods. The values in the table are collected from the original papers, if available, otherwise are computed by the authors. The abbreviations are: SPC – simple principal components (Vines, 2000), SPCA – sparse principal component analysis (Zou *et al.*, 2006), SCoTLASS – simplified component technique-LASSO (Jolliffe *et al.*, 2003) with $\boldsymbol{\tau} = (2.5\ 1.5\ 1.5\ 1.01\ 1.01\ 1.01)$, DSPCA – direct sparse PCA (d'Aspremont *et al.*, 2007), ESPCA – exact sparse PCA (Moghaddam *et al.*, 006b), SCA – simple component analysis (Rousson and Gasser, 2004) and IDR – interpretable dimensionality reduction (Chipman and Gu, 2005) with H, C, and S for homogeneity, contrasts and sparsity constraints respectively.

Table 3: *SC loadings and variances explained by different methods, Pitprop data*

Method	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	CV	CAV	0s
sBarse 1	-.41	-.41	0	0	0	0	-.41	-.41	-.41	-.41	0	0	0	29	29	7
sBarse 2	0	0	.71	.71	0	0	0	0	0	0	0	0	0	44	43	11
sBarse 3	0	0	0	0	.71	.71	0	0	0	0	0	0	0	54	53	11
SPC 1	.32	.32	0	0	.32	.32	.32	.32	.32	.32	0	-.32	-.32	28	28	3
SPC 2	.44	.44	.22	.22	-.44	0	0	0	.22	0	.22	.22	.44	47	46	4
SPC 3	.08	.08	-.37	-.37	-.21	-.46	-.37	.33	.04	.33	0	-.29	.12	61	59	1
SPCA 1	-.48	-.48	0	0	.18	0	-.25	-.34	-.42	-.40	0	0	0	28	28	6
SPCA 2	0	0	.79	.62	0	0	0	-.02	0	0	0	.01	0	42	42	9
SPCA 3	0	0	0	0	.64	.59	.49	0	0	0	0	0	-.02	57	55	9
SCoTLASS 1	-.48	-.49	0	0	0	-.11	-.38	-.25	-.38	-.41	0	0	0	30	30	6
SCoTLASS 2	0	0	.70	.71	0	.06	0	0	0	-.02	0	.01	0	45	44	8
SCoTLASS 3	-.06	-.09	-.02	0	.02	.22	.13	0	0	0	0	0	-.96	55	54	6
DSPCA 1	-.56	-.58	0	0	0	0	-.26	-.10	-.37	-.36	0	0	0	27	27	7
DSPCA 2	0	0	.71	.71	0	0	0	0	0	0	0	0	0	42	40	11
DSPCA 3	0	0	0	0	0	.79	.61	0	0	0	0	0	-.01	56	50	10
ESPCA 1	-.48	-.49	0	0	0	0	-.41	0	-.42	-.43	0	0	0	26	26	8
ESPCA 2	0	0	.71	.71	0	0	0	0	0	0	0	0	0	41	40	11
ESPCA 3	0	0	0	0	0	.81	.58	0	0	0	0	0	0	55	49	11
SCA 1	.45	.45	0	0	0	0	0	.45	.45	.45	0	0	0	25	25	8
SCA 2	0	0	0	0	.50	.50	.50	0	0	0	0	0	.50	35	34	9
SCA 3	0	0	.71	.71	0	0	0	0	0	0	0	0	0	49	47	11
IDR H1	-.38	-.38	0	0	0	-.38	-.38	-.38	-.38	-.38	0	0	0	30	30	6
IDR H2	-.30	-.30	-.30	-.30	.30	0	.30	.30	0	.30	-.30	-.30	-.30	47	46	2
IDR H3	-.33	-.33	0	.33	.33	.33	.33	-.33	-.33	0	0	0	-.33	61	57	4
IDR C1	-.15	-.15	-.15	-.15	-.15	-.15	-.15	-.15	-.15	-.15	.51	.51	.51	16	16	0
IDR C2	-.23	-.23	-.23	-.23	.40	0	.40	.40	0	.40	-.23	-.23	-.23	32	24	2
IDR C3	-.30	-.30	0	.37	.37	.37	.37	-.30	-.30	0	0	0	-.30	45	36	4
IDR S1	-.42	-.42	0	0	0	-.30	-.42	-.31	-.37	-.39	0	0	0	31	31	6
IDR S2	0	0	-.69	-.58	0	0	0	0	0	0	0	-.44	0	45	45	10
IDR S3	0	0	0	.43	.58	.57	0	0	0	0	0	0	-.39	59	56	9

The solutions produced by SPC (Vines, 2000), IDR H and C (Chipman and Gu, 2005) and SCoTLASS (Jolliffe *et al.*, 2003) are not sparse enough, and are not compared in the sequel. The worst solution seems to be the SCA one (Rousson and Gasser, 2004) which explains only 47% of the adjusted variance (and 66% for all six sparse components, also not much). The ESPCA solution (Moghaddam *et al.*, 006b) is the sparsest one but explains only 49% of the adjusted variance. DSPCA (d'Aspremont *et al.*, 2007) is a bit less sparse but also not quite satisfying with 50% adjusted variance. The sBarse solution is the sparsest one of the three remaining with explained 53% adjusted variance. The SPCA (Zou *et al.*, 2006) explains 55% adjusted variance for the price of 5 more non-zeros compared to the sBarse solutionis. The IDR solution (Chipman and Gu, 2005) with sparsity constraint (with $\eta = .9$) explains 56% adjusted variance being less sparse than the sBarse solution. However the IDR solution lacks orthogonality, which devaluates its quality as the sBarse and SPCA loadings are exactly orthonormal. Additional weakness of the IDR and SPCA solutions is that there are variables contributing to more than one SC. In fact, such overlapping effect is present in all solutions except the sBarse one. The sBarse solution of the Pitprop data seems to be the best one with respect to overall sparseness, ease of interpretation and goodness-of-fit.

The classical PCs are both orthogonal and uncorrelated. The sparse components cannot preserve these two features simultaneously. The orthogonality of the sparse components is maintained exactly only by the sBarse method, SCoTLASS and SPCA. The rest of the methods maintain the solutions' orthogonality only approximately, with the IDR (Chipman and Gu, 2005) deviating most. The correlations among the sparse components obtained by the three best sparse solutions of the Pitprop are given in Table 4. The correlation structures of the sBarse and SPCA solutions are quite similar.

Table 4: *Correlations among six SCs from three methods for the Pitprop data*

	sBarse $\alpha = .4$					SPCA					IDR Sparse ($\eta = .9$)				
Var	x_1	x_2	x_3	x_4	x_5	x_1	x_2	x_3	x_4	x_5	x_1	x_2	x_3	x_4	x_5
x_2	.16					-.17					.11				
x_3	-.26	-.19				-.33	.13				-.39	-.35			
x_4	.03	-.13	-.08			-.00	-.14	.10			-.26	.13	.17		
x_5	-.24	.20	.07	-.03		-.20	-.22	.14	.03		-.12	-.27	.09	-.05	
x_6	.15	-.07	-.33	.01	-.18	.08	.08	-.39	-.01	-.18	.09	-.08	.16	-.29	.07

Example 2 (The Gene expression data, $p \gg n$): The sBarse method can be applied on data sets where the number of variables (p) exceeds the number of samples (n). In gene expression data typically the number of genes (variables) are by far larger than the number of samples. Here, we use a real data set from gene expression measurements considered by Chin *et al.* (2006). The data are publicly available from http://icbp.lbl.gov/breastcancer/.

Witten *et al.* (2009) used this data set to illustrate the penalized matrix decomposition method for obtaining sparse principal components. They analyze 19,672 gene expression measurements on 89 samples. Due to computational reasons, Witten *et al.*

(2009) used subset of the data consisting 5% of the genes with highest variance. They computed the first 25 sparse principal components (SPC) and compared their result with sparse principal component analysis (SPCA) proposed by Zou *et al.* (2006). For comparison of the results, we also use the same subset of data, say $\mathbf{X}$, which consists of $p = 984$ genes (variables) and $n = 89$ samples.

The data are standardized and SVD is used to obtain the $p \times p$ matrix of principal component loadings, $\mathbf{A}_p$, and the corresponding matrix of singular values, $\mathbf{\Sigma}$. Since $p \gg n$, only a maximum number of n singular values are non-zero. Thus, the sBarse method is based on the $p \times n$ matrix of loadings $\mathbf{A}$ and the $n \times n$ diagonal matrix $\mathbf{\Lambda} = \mathbf{\Sigma}_n^2$.

The application of the sBarse method to the gene expression data set resulted in 88 sBarse components each having at least one non-zero loading (coefficient) of the genes. The number of non-zero loadings per sparse component ranges from 1 to 99, with a median number of 7. The number generally decreases with decreasing variances of the sBarse components. For instance, the first, second, and third sBarse components contain, respectively, 99, 83, and 89 genes, and each component is non-overlapping with respect to the genes.

We compare the sBarse method with SPC and SPCA methods, as demonstrated in Witten *et al.* (2009). The comparison is based on the level of sparsity of the components and the proportion of cumulative variances explained by the sparse components. The `SPC` function in `R` uses as one of its required argument the sum of absolute values (`sumabsv`) of loadings in a sparse component. This value is assumed to measure the level of sparsity and is set by the user. For comparison, we set two different values for `sumabsv`. The first is the maximum of the sum of the absolute values of the sBarse component loadings, and the second is the average of the sum of the absolute values of the sBarse component loadings. For the gene expression data under consideration, these values are found to be 9.9499 and 2.878, respectively. The top two plots in Figure 1 are based on `sumabsv` $= 9.9499$, and the bottom two – on `sumabsv` $= 2.878$.

The two plots on the left hand side of Figure 1 depict the number of non-zero loadings in each of the first 25 sparse components when `sumabsv` $= 9.9499$ and `sumabsv` $= 2.878$

for SPC (note that the results for SPCA are adjusted based on the SPC values, as explained in Witten *et al.* (2009)). The average number of nonzero loadings in a sBarse component is 26. For the SPC, this number is 188 when `sumabsv` $= 9.9499$ and 14 when `sumabsv` $= 2.878$.

The two right hand side plots show the corresponding proportion of cumulative variances explained by the sparse components when the two values of `sumabsv` are used.

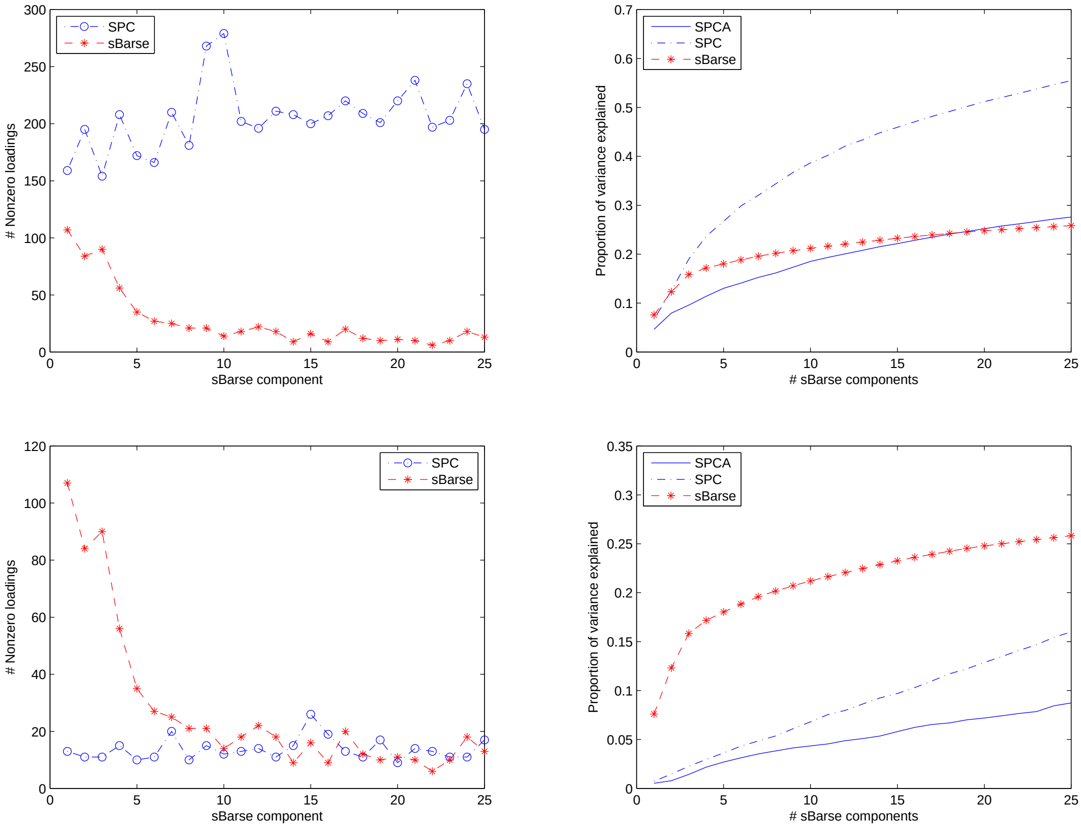

Figure 1: Comparison of sBarse, SPC and SPCA with respect to sparsity (Left) and the cumulative percentage of variance explained (Right). The two upper plots are based on `sumabsv` $=$ max(sum(abs(sBarse))) $= 9.9499$ for SPC and the two lower plots are based on `sumabsv` $=$ mean(sum(abs(sBarse))) $= 2.878$.

The two top plots show that the sBarse components are sparser than the SPCs, but explain lower percentage of cumulative variance. However, the SPCA, having the same number of non-zero loadings as the SPC, explains lower cumulative percentage than the sBarse components for the first 25 sparse components. Thus, the sBarse method outperforms the SPCA method both in the level of sparsity and the cumulative percentage of variance explained. Note, that the sBarse method is based on the adjusted variance suggested by Zou *et al.* (2006).

From the two lower plots, where the sparsity of the SPC is allowed to increase (i.e., getting sparser), the resulting sparse components explain a smaller proportion of the variances. In this case, the proportion of variances explained by the sBarse components exceeded that of SPCs. The nonzero loadings for the first few sBarse components are larger than the SPCs, but gradually becomes similar.

In general, the sBarse method tends to produce sparser components than the SPC method but explaining lower cumulative percentage of variation. On the other hand, the SPC method tends to produce components explaining higher cumulative percentage of variation than the sBarse method but with less sparseness for the components. However, the higher percentage of explained variation by the SPCs may be attributed to the fact that the components are not orthogonal. Note that the sBarse components are orthogonal to each other by construction.

Each sBarse component contains cluster of genes, which are not contained in the other sBarse components. Such non-overlapping characteristics in the contents of the sparse components can facilitate the interpretation of the components. This behavior is not common in the other sparse methods, and particularly in SPC.

The computation time normally depends on the size of the data set, the software used, the processor of the computer, the specific values used in the user-defined functions, and so on. The SPC method is based on `R-2.9.0` (using the PMA package), while the sBarse method is programmed in `MATLAB`. As a result direct comparison of the methods may not be feasible. But, the sBarse method is reasonably fast for analyzing this gene expression data set.

The size of the step length in searching for the best $\alpha \in [0, 1]$ can affect the computational time, especially for large data, as the gene expression data set. In general, the smaller step length guarantees better solution, but it takes longer time. Hence, it is recommended to consider a compromise between the computation time, the required level of sparsity, and the total variance explained while choosing the step length.

In fact, the computation time can be reduced considerably by narrowing the search interval(s) around the best sBarse solution(s) found in the previous search interval(s). For the gene expression data set, analyzed on an Intel Pentium 4(3.20GHz, 0.99GB of RAM) desk top computer, it takes 4.14 minutes to get the best solution with $\alpha = .15$. The value of α is obtained by successively narrowing the search interval at each step, leading to a decrease in the step lengths. The solution contains 88 sBarse components, i.e. 88 orthogonal and non-overlapping combinations of genes. This is .09% of the original dimension of the gene expression data (984 genes), i.e. dimensionality reduction of more than ten times.

The adjusted variance of the first six sBarse components is 18.2%. The corresponding values for SPC (when `sumabsv` = 9.9499) and SPCA are, respectively, 18.9% and 12.5%. These percentages become 3.4% and 2.6%, respectively, when `sumabsv` = 2.878 is used. On the other hand, the number of genes in the first six sBarse clusters (that is, the number of non-zero loadings for the first six sBarse components) are 99, 83, 89, 55, 35, and 27, respectively, accounting for 39.4% of all 984 genes. For the SPC with `sumabsv` = 9.9499, the number of non-zero loadings in the first six SPCs are 149, 169, 122, 191, 152, and 155. These numbers become 13, 12, 11, 15, 10, and 11 when `sumabsv` = 2.878. Unlike the sBarse components, both the SPC and SPCA produce components involving overlapping subsets of genes.

5 Discussion and Conclusion

The paper proposes a computationally efficient method that simplifies the interpretation of PCs, with several advantages over the existing ones. The method is designed in such a way that the variables associated with each sBarse component do not overlap, leading to clearer interpretation of the components. The elements of the vector of loadings for a resulting component take values from $\{0, \pm c\}$. The first m sBarse components are used for interpretation as the rest $p - m$ ones are identically zero. Hence, the method automatically suggests the number of components to retain that accounts for the majority of the information, as these last components do not account for the total variation. This feature can be used as an alternative to the existing methods as the scree plot, and the cumulative percentage of variances which both involve subjective judgment.

The examples illustrate that the sBarse method performs at least as well as the other similar sparse component methods. It gives the sparsest orthogonal components compared to the other similar methods. In addition, it can be readily applied to data sets with $p \gg n$. For the gene expression data set, the method results in orthogonal sBarse components which are non-overlapping with respect to the genes. Thus, the sBarse method for such data set helps to cluster the genes in such a way that the genes forming the first few clusters (sBarse components) explain the majority of the information in the original data and each cluster of genes contain an independent information which is not captured by the other clusters of genes. This way, the method helps to reduce the dimensionality of the data. Some clusters contain a single gene which means that this gene may contain good information by itself and worth further investigation.

One problem of the sBarse components approach is that one may be unable to produce sparse loadings for any $m = 1, \ldots, p$. However, this is not a serous problem as we are interested in a minimal number of components with orthogonal loadings matrix, and accounting for a maximal percentage of total variation.

Another problem might be the lack of clear guidance on the number of α's to consider in order not to skip the best sparse solution. But, taking only few of them suffices, as intervals of α's correspond to the same sBarse solution.

References

Anaya-Izquierdo, K., Critchley, F., and Vines, K. (2008). Orthogonal simple component analysis. *Technical report*, **08/11**, http://statistics.open.ac.uk/TechnicalReports.htm.

Cadima, J. and Jolliffe, I. T. (1995). Loadings and correlations in the interpretation of principal components. *Journal of Applied Statistics*, **22**, 203–214.

Cadima, J. and Jolliffe, I. T. (2001). Variable selection and the interpretation of principal subspaces. *Journal of Agricultural, Biological, and Environmental Statistics*, **6**, 62–79.

Chin, K., Devries, S., Fridlyand, J., Spellman, P., Roydasgupta, R., Kuo, W., Lapuk, A., Neve, R., Qian, Z., Ryder, T., Chen, F., Feiler, H., Tokuyasu, T., Kingsley, C., Dairkee, S., Meng, Z., Chew, K., Pinkel, D., Jain, A., Ljung, B., Esserman, L., Albertson, D., Waldman, F., and Gray, J. (2006). Genomic and transcriptional aberrations linked to breast cancer pathophysiologies. *Cancer cell*, **10**, 529–541.

Chipman, H. A. and Gu, H. (2005). Interpretable dimension reduction. *Journal of Applied Statistics*, **32**, 969–987.

d'Aspremont, A., Ghaoui, L., Jordan, M., and Lanckriet, G. (2007). A direct formulation for sparse pca using semidefinite programming. *SIAM Review*, **49**, 434–448.

d'Aspremont, A., Bach, F., and Ghaoui, L. (2008). Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, **9**, 1269–1294.

Farcomeni, A. (2009). An exact approach to sparse principal component analysis. *page to appear.*

Gabriel, K. R. (1971). The biplot graphic display of matrices with application to principal component analysis. *Biometrika*, **58**, 453–467.

Gabriel, K. R. (2002). Goodness of fit of biplots and correspondence analysis. *Biometrika*, **89**, 423–436.

Gower, J. C. and Hand, D. J. (1996). *Biplots*. Chapman & Hall, London.

Hausman, R. E. (1982). Constrained multivariate analysis. In S. H. Zanakis and J. S. Rustagi, editors, *Optimization in statistics*, pages 137–151. North-Holland, Amsterdam.

Jeffers, J. N. R. (1967). Two case studies in the application of principal component analysis. *Applied Statistics*, **16**, 225–236.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer-verlag, New York.

Jolliffe, I. T., Trendafilov, N. T., and Uddin, M. (2003). A modified principal component technique based on the lasso. *Journal of Computational and Graphical Statistics*, **12**, 531–547.

Journée, M., Nesterov, Y., Richtárik, P., and Sepulchre, R. (2008). Generalized power method for sparse principal component analysis. Technical Report 2008/70. http://www.uclouvain.be/en-44508.html.

Lykou, A. and Whittaker, J. (2009). Sparse cca using a lasso with positivity constraints. *Computational Statistics and Data Analysis*, page to appear.

Moghaddam, B., Weiss, Y., and Avidan, S. (2006a). Generalized spectral bounds for sparse lda. In *Proceedings of the 23rd international conference on Machine learning*, Pittsburg. PA.

Moghaddam, B., Weiss, Y., and Avidan, S. (2006b). Spectral bounds for sparse pca: Exact and greedy algorithms. *Advances in Neural Information Processing Systems*, **18**, 915–922.

Robert, P. and Escoufier, Y. (1976). A unifying tool for linear multivariate statistical methods: The rv-coefficient. *Applied Statistics*, **25**, 257–265.

Rousson, V. and Gasser, T. (2004). Simple component analysis. *Applied Statistics*, **53**, 539–555.

Sabatier, R. and Reynès, C. (2008). Extensions of simple component analysis and simple linear discriminant analysis using genetic algorithms. *Computational Statistics and Data Analysis*, **52**, 4779–4789.

Seber, G. A. F. (2004). *Multivariate Observations*. Wiley, New Jersey.

Trendafilov, N. I. and Jolliffe, I. T. (2007). Dalass: Variable selection in discriminant analysis via the lasso. *Computational Statistics and Data Analysis*, **51**, 3718–3736.

Trendafilov, N. I. and Vines, K. (2009). Simple and interpretable discrimination. *Computational Statistics and Data Analysis*, **53**, 979–989.

Vines, S. K. (2000). Simple principal components. *Applied Statistics*, **49**, 441–451.

Witten, D. M., Tibshirani, R., and Hastie, T. (2009). A penalizex matrix decomposition, with applications to sparse principal components and canonical correlation. *Biostatistics*, **10**, 515–534.

Zou, H., Hastie, T., and Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, **15**, 265–286.