Sparse PCA for gearbox diagnostics

Anna Bartkowiak
Institute of Computer Science, University of Wrocław, and Wrocław High School of Applied Informatics
Wrocław, Poland
Email: aba@ii.uni.wroc.pl

Radosław Zimroz
Wrocław University of Technology
Vibro-Acoustic Science Laboratory
Wrocław, Poland
Email: Radoslaw.Zimroz@pwr.wroc.pl

Abstract—The paper presents our experience in using sparse principal component (PCs) (Zou, Hastie and Tibshirani, 2006) for visualization of gearbox diagnostic data recorded for two bucket wheel field excavators, one in bad and the other in good state. The analyzed data had 15 basic variables. Our result is that two sparse PCs, based on 4 basic variables, yield similar display as classical pair of first two PCs using all fifteen basic variables. Visualization of the data in Kohonen’s SOMs confirms the conjecture that smaller number of variables reproduces quite well the overall structure of the data. Specificities of the applied spca method are discussed.

I. INTRODUCTION

In our times we are witnessing a growing interest in the predictive assessment of industrial machinery. Machines work everywhere and are critical in maintenance of environmental and life conditions of humans. Machines use gearboxes, which in turn are substantial for proper functioning of the devices. Bad functioning is connected with huge economic losses; therefore early detection of malfunctioning is crucial both from economic and vital aspect. Although there are many general rules how to assess the problem, the working devices are quite different and different approaches are needed. What concerns machinery faults, vibration analysis has become almost the universal method to assess the state of a machine. See [12], [1], [2], [3], [11] for methods used in this domain.

We will be concerned with diagnosis of gearboxes, used in bucket wheel excavators, working in surface mining. These are huge and costly machines. In the following we will analyze vibration sounds obtained from two machines: one in bad state (machine A), and one in good state (machine B). We will show that, on the basis of vibration sounds, it is possible to assess, what is the state of the machine: good or bad. This will be done by two combined methods: using self-organizing maps and sparse principal components. Next Section describes how the data were acquired, and gives some preliminary characterization of the recorded data. A preliminary approach to dimensionality reduction - by using Kohonen’s self-organizing maps (SOMs) is shown in Section 3. The principles of constructing classical and sparse principal components are presented in Section 4. Results of applying sparse PCA to the recorded data, and discussion on the results may be found in Section 5. An overall discussion on the applied methods and their validity is presented in Section 6, where also a short summary of the results may be found.

II. THE DATA

A. Collection of the data

Data were recorded in an experiment carried out in the Vibro-Acoustics Science Laboratory of Wroclaw University of Technology. Two complex multistage gearboxes implanted in two bucket wheel drive units were investigated. The Bruel&Kjaer pulse system was used in the experiments. The vibration signals were measured in four vibration channels. The signal duration was 60s and the sampling frequency 16384 Hz. After data acquisition, the speed profile and the vibration signal data were processed. This was done by (a) signal segmentation (according to digging process cyclicity), and (b) feature extraction. The preprocessing yielded the mean speed for each segment obtained from the speed profile and 15 amplitudes of the planetary mesh frequency components. Some elements of the processing are depicted in Fig. 1.

* This work was partially supported by (Polish) State Committee for Scientific Research in 2009 - 2012 as research project by R. Zimroz.
The recorded vibration signal is shown as two graphs in the top exhibit of the Figure, depicting speed [RPM] and acc [m/sec²] (acceleration) as function of time [sec]. To achieve stationarity, the recorded signal was cut into segments (two of them are shown in the top exhibit of Figure 1). Each of the obtained segments was subjected to fft (Fast Fourier Transform) analysis using Matlab function psd. This yielded for each segment a power spectrum. Two such power spectra are shown in Fig. 1, bottom exhibit. One may notice some periodicity in appearing of the peaks of the spectrum: all they appear at equally spaced values of mesh frequency (expressed in Hz). This can be explained, taking into account the architecture of the gearbox and its modus of work (to learn more, see [1], [2], where also a preliminary analysis of the data may be found).

As result of this stage of data recording and preprocessing, two sets of data vectors were obtained, each with 15 components. Each data vector corresponds to the recorded variables pp1, . . . , pp15 obtained from the fft analysis carried out for one segment of the vibration recording. The obtained data vectors were recorded in two matrices of size \([n_A \times 15]\) and \([n_B \times 15]\), with \(n_A = 1232\) and \(n_B = 951\). These matrices constitute the basis of our further analysis.

Why just 15 power spectra were recorded? This was not clearly defined. Generally, the obtained variables pp1, . . . , pp15 are expected to indicate whether the device is in bad or good state; they are not expected to be associated with correct state of particular elements of the working device. What concerns the gearbox A, it was found that all the rolling bearings had exceeded the allowable radial backlash and most of the gears had scuffs and microcracks on their teeth [2].

### B. Preliminary analysis of the data: simple statistics and visualization

In first place univariate characteristics such as means, medians and boxplots were calculated. This was done both for raw and normalized data. Each matrix \(X\) has been firstly centered to have the mean of each column equal zero, and next normalized so to have the length of each column equal to 1 (this was the way used by [10] and [8] in their sparse PCA). Then every column of \(X\) has variance equal to \(1/n\). The boxplots obtained for both raw and normalized data matrices are shown in Fig. 2.

Looking at the boxplots constructed from raw data of sets A and B, one may notice that their distributions differ (in absolute value) considerably in these two sets. For example, in set A, the most characteristic are very large values in variable 5 and 7; while in set B the largest values are exhibited by variables 2 (contains exceedingly large part of outliers) and variable 4.

On the other hand, looking at the variables no. 9–15, which have very small amplitudes, but differentiated distributions, one is not sure whether, when expressing these variables in magnified scale, a better differentiation between group A and group B might be found.

---

**Fig. 2.** The top two boxplots depict groups A (bad) and B (good) with raw data, i.e. non-standardized. The bottom two exhibits depict set C containing the sets A and B merged into one common group. Two boxplots are shown for the set C: firstly for raw data, next for normalized data.
The medians calculated for all the 15 variables are shown in Fig. 3. There are two plots constructed for raw and standardized data of both groups. The bottom plot, exhibiting medians for normalized data, is especially interesting. One may notice that:

(i) variable no. 1 has practically the same value in both groups, therefore it is hard to expect that this variable will contribute to the differentiation between groups A and B; the same can be said about variable no. 13;

(ii) variable no. 2 has inverted value as compared to remaining variables; what does it mean? a question for the specialists;

(iii) remaining variables No.s 3 – 12, 14 – 15 exhibit approximately the same difference between their medians, thus any one of them is a candidate for a good discriminator.

Of course, this is only a preliminary reasoning, to be more strict, one should consider a measure of variability (variance) of successive medians.

III. MULTIVARIATE DATA VISUALIZATION USING KOHONEN’S SELF-ORGANIZING MAPS

The question to answer: Have the variables pp1 - pp15 some discriminative power with respect to the bad and good state of the machine? The problem was considered in [12] by using canonical discriminant analysis applied using all the 15 variables, and the answer is positive. So other two questions:

Is Kohonen’s SOM suitable to show the difference between set A and B indicating a bad and a good state of the machine? If yes, can the set of 15 variables be somehow reduced, without losing its indicative properties of the state of the machine?

For construction of the maps we use the Matlab SOM Toolbox [9]. The architecture of the map is hexagonal; the map is composed from $24 \times 10$ hexagons. The data were standardized statistically to have zero mean and unit variance for each column. In Fig. 4 we show two maps, one when taking all 15 variables, the other based on variables No. 2-8.

Looking at the displayed maps, one may notice that both maps differentiate quite nice data from group A and group B. There are together $24 \times 10$ codebook vectors located at centers of the hexagons. In the upper map only two of them (no. 217 and 218) have mixed group content, i.e. represent data vectors belonging either to A or to B.

In the bottom map, the topological locations are inverted with respect to the y-axis (this happens quite often, both
maps are topologically equivalent). In this map also only two hexagons (no. 1 and 2) have mixed group content, the others are perfectly disjoint. Apart from this, both maps show a decided separation of hexagons containing either bad (A) or good (B) data.

The quality of the maps is measured by two indices: \( q \) - quantization error and \( t \) - topographic error (see [9] for definitions). The respective errors amount:

| when using all variables: | \( q = 1.075 \), \( t = 0.041 \) |
| when using variables 2–8: | \( q = 0.586 \), \( t = 0.040 \) |

Guided by the considerations in previous section when inspecting the plot of normalized medians, we have made other searches; for lack of space we do not discuss them here. A principled way for finding relevant variables is presented in next section.

IV. ORDINARY AND SPARSE PRINCIPAL COMPONENTS

A. The classical PCA

Principal component analysis (PCA) is one of the most common and widespread methods for multivariate linear data analysis. It serves for investigating data structure, data mining, data smoothing and approximation, also for exploring data dimensionality. The method permits to build new features, called principal components (PCs), which may serve for visualization of the data [7].

Let \( \mathbf{X} \) of size \( n \times d \) denote the observed data matrix. For simplicity of presentation, assume that \( n > d \) and that \( \mathbf{X} \) is of full rank. It is advised to standardize or normalize the matrix \( \mathbf{X} \). Following [10], [8], we assume that the data matrix is column wise centered and has columns of unit length. This means that all columns have means = 0 and variances equal \( 1/n \). The PCA starts from computing the eigenvalues (\( \lambda \)) and eigenvectors (\( \mathbf{v} \)) of the cross-product matrix \( \mathbf{S} = \mathbf{X}^\mathsf{T} \mathbf{X} \) satisfying the matrix equation \( (\mathbf{S} - \lambda \mathbf{I}) \mathbf{v} = \mathbf{0} \). This results in \( d \) eigenvalues

\[
\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d
\]

and associated with them \( d \) eigenvectors

\[
\mathbf{v}_j = (v_{1j}, \ldots, v_{dj})^\mathsf{T}, \quad j = 1, \ldots, d.
\]

The eigenvectors constitute the loading matrix \( \mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_d] \). The two fundamental PCA paradigms are:

1) Feature construction:

\[
\mathbf{Z}^{(K)}_{n \times K} = \mathbf{X} \ast [\mathbf{v}_1, \ldots, \mathbf{v}_K], \quad 1 \leq K \leq d.
\] (3)

The new features identified as columns of \( \mathbf{Z}^{(K)} \) are called Principal Components (PCs).

2) Data reconstruction:

\[
\hat{\mathbf{X}}^{(K)}_{n \times d} = \mathbf{Z}^{(K)} \ast ([\mathbf{v}_1, \ldots, \mathbf{v}_K]^\mathsf{T}).
\] (4)

Taking \( K = d \), the full original data matrix \( \mathbf{X}_{n \times d} \) is reconstructed. For \( K < d \) the best linear approximation of \( \mathbf{X}_{n \times d} \) by a rank-\( K \) matrix \( \hat{\mathbf{X}}^{(K)}_{n \times d} \) is obtained – it is best in the meaning of the L2 norm.

The constructed PCs have the major advantage that they are uncorrelated, which permits to analyze each of them separately, without referring to the others.

Principal components can be also computed via SVD:

\[
\mathbf{X} = \mathbf{U} \mathbf{D} \mathbf{V}^\mathsf{T}
\]

where \( \mathbf{Z} = \mathbf{U} \mathbf{D} \) contains the PCs, and \( \mathbf{V} \) the loadings.

A detailed analysis of the considered data (sets A, B, C) is shown in [12]. Among others it was found that the first three PCs calculated from set C explain 86 % of total variance of the data. Displaying a scatterplot of the first two PCs (i.e. PC1 and PC2) the points-projections of sets A and B are practically separated and it is possible to draw a linear line separating the two sets with only 5 misclassified data points (out of total \( n_A = 1232 \) and \( n_B = 951 \) data points).

B. Sparse PCA

The idea of sparse PCA: To construct a PC, one needs all the variables contained in the data set. Is this necessary? It would be desirable to reduce the number of explicitly used variables. This idea was floating for some time among the data analysts and some proposals were offered (see [7], [8], [10]). A reasonable approach with a fast implementation algorithm was proposed in [10]. The authors proposed a regressive approach. The idea starts from the observation that each PC is a linear combination of all \( d \) variables. Thus, for known values of a given PC (given as vector \( z \), being a column of the matrix \( \mathbf{Z} \) defined in formula 3), and for known values of the matrix \( \mathbf{X} \), we are able to recover the coefficients of that combination by applying a regression method. If we want a sparse PC (which means that it is based only on few original variables), we should apply here a sparse regression method.

The authors of [10] proposed for this purpose the sparse elastic net regression [5], [8].

The computational algorithm proposed by Zou, Hastie et Tibshirani: The sparse PCs are computed in an iterative way. The computations are for fixed \( K \) (\( 1 \leq K \leq d \)) chosen by the user. The starting point of the reasoning is the observation that the eigenvectors \( \mathbf{v}_j, \quad j = 1, \ldots, \hat{K} \) constituting the matrix \( \mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}^{(K)}] \) appear in PCA in two roles: when constructing the features (formula 3)

\[
\mathbf{Z}^{(K)} = \mathbf{XV}^{(K)},
\] (5)

or reconstructing the original data matrix (formula 4).

\[
\hat{\mathbf{X}}^{(K)} = \mathbf{Z}^{(K)} \ast ([\mathbf{v}_1, \ldots, \mathbf{v}^{(K)}]^\mathsf{T}).
\] (6)

Let \( \mathbf{B} \) denote the matrix \( \mathbf{V}^{(K)} \) used in construction of \( \mathbf{Z}^{(K)} \) (formula 5), and \( \mathbf{A} \) the matrix \( \mathbf{V}^{(K)} \) used in reconstruction of \( \hat{\mathbf{X}} \) (formula 6). To obtain sparse PCs, the authors [10] propose to estimate the loading matrix \( \mathbf{V}^{(K)} \) in an alternating way, working in two stages:

Stage 1. Given \( \mathbf{A} \), compute \( \mathbf{B} \);

Stage 2. Given \( \mathbf{B} \), compute \( \mathbf{A} \).

The initial estimate of \( \mathbf{A} \) may be obtained, e.g. by ordinary PCA.
Stage 1. Given A, compute B. Having A, we may perform for each \( j = 1, \ldots, K \) the following evaluations:

- compute the values (scores) of the \( j \text{th} \) feature \( z_j \) appearing in the \( j \text{th} \) column of \( Z^{(K)} \) as:
  \[
z_j = X * a_j,
\]
- notice that the scores \( z_j \) may be also computed as
  \[
z_j = X * b_j, \quad (7)
\]
- notice, that at this moment we know \( z_j^* \), and so we may substitute in equation \( (7) \): \( z_j = z_j^* \), obtaining
  \[
z_j^* = X * b_j, \quad (8)
\]
- since equation \( (8) \) may be viewed as a linear regression problem in unknown regression coefficients vector \( b_j \), obtain an estimate of \( b_j \) by applying a sparse regression algorithm (e.g larsen [5]).

In such way it is enforced that only few columns of \( X \) will appear in the regression equation, because only few elements of the vector \( b \) will have non-zero elements. The degree of sparseness of the regression depends on the algorithm. The larsen algorithm permits the user to declare, how many original variables (columns of \( X \)) should be retained.

Performing the estimation of sparse regression coefficients vectors \( b_j \) for \( j = 1, \ldots, K \) we obtain \( K \) successive vectors composing estimates of the sought matrix \( \hat{B} = [b_1, \ldots, b_K] \) – which was desired at this stage of the calculations.

Now we pass to stage 2.

Stage 2. Given B, compute A.

If \( B \) is fixed, then the problem is to find such a matrix \( A \), that minimizes the quadratic form

\[
\|X - (XB)A^T\|^2
\]

subject to the restriction that \( A^T A = I_{K \times K} \).

It is shown in [10] that this is obtained by a reduced rank form of the Procrustes rotation by computing the SVD of \( (X^T X)B \):

\[
(X^T X)B = UDV^T
\]

and substituting \( A = UV^T \).

Stages 1 and 2 are repeated alternately until a final criterion of convergence is met. The convergence might be achieved if the maximal difference computed for the respective elements \( B_{ik} (i = 1, \ldots, d, k = 1, \ldots, K) \) is smaller than an assumed small number \( \epsilon \) (say, \( \epsilon = 10^{-6} \))

\[
|B^{old}_{ik} - B^{new}_{ik}| < \epsilon,
\]

where \( B^{old} \) and \( B^{old} \) denote the matrices of loadings obtained in two successive iterations. This condition might be combined with another condition that the total number of iterations carried out so far is smaller than a declared maximal number of iterations.

V. Sparse PCA – results for the gearbox data

For calculations we have used the Matlab software implemented by Karl Skoglund and available at http://www2.imm.dtu.dk/~ksjo/kas/software/index.html [8]. Many thanks to him! Assuming \( K=3, stop=2 \) (each PC needs only 2 original variables for its construction) and \( lambda=1 \) (\( lambda \) is a parameter needed by the function larsen called in Stage 1 of the algorithm) we got for our data the (constructed) feature matrix \( Z_{3 \times 3} \) containing in its columns three sparse principal components: \(<\text{PC1, PC2, PC3}>\). They are depicted in Fig. 5.

The displays look much similar to those obtained by or-
ordinary PCA (shown in [12]). In our calculations we have assumed $K=3$, $stop=2$ (each PC uses only two original variables) and $\text{lambda}=1$ (a parameter needed by the function \text{larsen} called in stage 1 of the spca algorithm). Similar displays (not shown here) were obtained when using $\text{lambda}=10$ and $\text{lambda}=100$.

When analyzing the results, one should also consider:

- Which original variables were chosen by the applied spca algorithm to constitute the sparse principal components?
- How much of total variance is explained by the constructed sparse PCs?

Details of the results, for the same values of $K (=3)$, $stop (=2)$ but different values of $\text{lambda}$ are shown in Table 1.

Before discussing the results, we should emphasize that the applied method yields sparse PCs that have different properties than the classic ones. This happens because the classical PCs are set with the criterion of extract the maximum amount of total variance of the data, while in construction of sparse PCs this principle is \textit{de facto} not used. To say it plainly, the columns of $\mathbf{B}$ (used for construction of $\mathbf{Z}$) are based on other principle: the sparse regression. This makes that the variances of the sparse PCs need not decrease with the no. of the PC; also that the variances of the sparse PCs do not need to sum to the total variance, given as \textit{trace} of $\mathbf{S} = \mathbf{X}^T \mathbf{X}$. To make them act similarly like the classical PCs, after convergence of the 2-stage algorithm the following actions are undertaken:

- normalization of the loadings appearing in $\mathbf{B}$, to have unit Euclidean length.
- ordering of the columns of the $\mathbf{Z}$ matrix according to their decreasing variance, and next orthogonalizing them – to have adjusted increments (AV) of explained variance.

Now let us look at Table 1. We find that:

- Considering point (i) formulated above: The variables chosen for two pairs of the sparse PCs are the same for $\text{lambda}=1$ and $\text{lambda}=2$, but one pair of chosen variables differs.
- Considering point (ii) formulated above: An inversion happened in finding the PCs with the largest variances. In the experiment using 3 different values of $\text{lambda}$, the sparse PC explaining the maximum variance was found as the third (for $\text{lambda}=1$) or the second (for $\text{lambda}=10$ and $\text{lambda}=100$).

- Generally, the total percentage of explained variance called $AV$ (adjusted variance) is low, however that the graphically displayed information of the data is practically the same as that obtained from first three classical PCs, which reconstruct together 86.05% of total variance. This leads to the conclusion that the spca is able to recover the proper information on the topological structure of the data. This is interesting and needs further elaboration.

VI. DISCUSSION AND CONCLUDING REMARKS

The sparse algorithm proposed in [10] is mathematically very sophisticated. It can be implemented to work effectively and fast when using Matlab [8]. We have applied this algorithm with the aim of reducing the number of variables needed when making a diagnosis of gearbox functioning. It seems that we were successful and the set of used variables can be reduced. However, we have noticed that the results of analysis depend from few initial parameters needed by the spca algorithm. For the moment the parameters are found by a kind of cross-validation. The results may also depend on the structure of the data. Therefore, before advising the algorithm for an automated procedure, some further research on work of the algorithm is needed.

REFERENCES