Lesson 5

Tackling with Redundancy

The collinearity problem is essentially caused by redundancy in the data: Measurements are more or less dependent of each other. However, none of the measurements is *completely* useless, each of them typically delivers *some* fresh information. Qualitative analyses cannot help here — on the other hand, when the quantitative approach is adopted, powerful methods turn out to be readily available.

5.1 Some linear algebra

Linear algebra is a highly abstract field of systems theory. In this context, it suffices to concentrate on just a few central ideas, and theoretical discussions are kept in minimum; these issues are studied in more detail, for example, in [15] or [33].

5.1.1 On spaces and bases

To have a deeper understanding of how the mapping from the "space" of input variables into the "space" of output variables can be analyzed, basic knowledge of linear algebra is needed. The main concepts are *space, subspace,* and *basis.* The definitions are briefly summarized below:

> The set of all possible real-valued vectors x of dimension n constitutes the *linear space* $\mathcal{R}^n$. If $\mathcal{S} \in \mathcal{R}^n$ is a set of vectors, a *subspace* spanned by $\mathcal{S}$, or $\mathcal{L}(\mathcal{S})$, is the set of all linear combinations of the vectors in $\mathcal{S}$. An (ordered) set of linearly independent vectors θ_i spanning a subspace is called a *basis* for that subspace.

Geometrically speaking, subspaces in the n dimensional space are hyperplanes (lines, planes, etc.) that go through the origin. The number of linearly independent vectors in the subspace basis determines the *dimension* of the subspace. The basis vectors θ_1 to θ_N can conveniently be represented in a matrix form:

$$\theta = \left(\begin{array}{c|c|c} \theta_1 & \cdots & \theta_N \end{array} \right). \tag{5.1}$$

This basis matrix has dimension $n \times N$, assuming that the dimension of the subspace in the n dimensional space is N. Given a basis, all points x in that subspace have a unique representation; the basis vectors θ_i can be interpreted as coordinate axes in the subspace, and the "weights" of the basis vectors, denoted now z_i, determine the corresponding "coordinate values" (or *scores*) of the point:

$$x = \sum_{i=1}^{N} z_i \cdot \theta_i. \tag{5.2}$$

The elements in θ_i are called *loadings* of the corresponding variables. In matrix form, the above expression can be written as

$$x = \theta \cdot z. \tag{5.3}$$

Further, if there are various data vectors, the matrix formulation can be written as

$$X = Z \cdot \theta^T. \tag{5.4}$$

There is an infinite number of ways of choosing the basis vectors for a (sub)space. One basis of special relevance is the so called "natural" basis: fundamentally, all other bases are defined with respect to this natural basis. For the space of n measurements the natural basis vector directions are determined directly by the measurement variables; formally speaking, each entry in the data vector can be interpreted as a coordinate value, the basis vectors constituting an identity matrix, $\theta = I_n$.

However, even though this trivial basis is easy to use, it is not necessarily mathematically the best representation for the data (as was shown in the example about collinearity above). Next we see how to change the basis.

5.1.2 About linear mappings

The matrix data structure has been adopted here for various purposes — this is partly duw to the role of `Matlab` as the assumed basic tool: There (at least originally) the matrix was the only one data structure available. The matrix can have various roles. It can be used as a collection of data values (as X and Y above, for example), or it can be used as a frame for a vector system (as in the case of basis vectors); but perhaps the most important role of a matrix is its use as a means of accomplishing *linear transformations* between different bases of (sub)spaces.

Whereas all matrix operations can be interpreted as linear transformations, now we are specially interested in mappings between different bases. The transformations from a given basis to the natural basis are straightforward: applying (5.3) gives the transformed coordinates directly. The question that arises is how one can find the coordinate values z for a given x when the new basis θ is given. There are three possibilities depending on the dimensions n and N:

- If $n \equiv N$, matrix θ is square and invertible (because the linear independence of the basis vectors was assumed). Then one can directly solve

$$z = \theta^{-1} \cdot x. \tag{5.5}$$

- If $n > N$, the data point cannot necessarily be represented in the new basis. Using the least squares technique (see the first lesson) results in an approximation

$$\hat{z} = \left(\theta^T \theta\right)^{-1} \theta^T \cdot x. \tag{5.6}$$

- If $n < N$, there are an infinite number of exact ways of representing the data point in the new basis. Again, the least squares method offers the solution, now in the sense of minimizing $z^T z$, that is, finding the minimum numerical values of the coordinates (see page 20):

$$z = \theta^T \left(\theta \theta^T\right)^{-1} \cdot x. \tag{5.7}$$

All of the above cases can be conveniently presented using the *pseudoinverse* notation:

$$z = \theta^{\dagger} \cdot x. \tag{5.8}$$

If the basis vectors are *orthonormal* (orthogonal and normalized at the same time, meaning that $\theta_i^T \theta_j = 0$, if $i \neq j$, and $\theta_i^T \theta_j = 1$, if $i = j$) there holds $\theta^T \theta = I_N$ (or $\theta \theta^T = I_n$, whichever is appropriate). Thus, all the above formulas (5.5), (5.6), and (5.7) give a very simple solution:

$$z = \theta^T \cdot x, \tag{5.9}$$

or, corresponding to (5.4),

$$Z = X \cdot \theta. \tag{5.10}$$

The above result visualizes the benefits of basis orthonormality; there are additional advantages that are related to the numerical properties of orthogonal transformation matrices (manipulations in an orthonormal basis are optimally conditioned)[1].

5.1.3 Data model revisited

To enhance the basic regression method, a more sophisticated scheme is now adopted (see Fig. 5.1). Speaking informally, we search for an "internal structure" that would capture the system behavior optimally; this internal structure is assumed to be implemented as a linear subspace. The data samples are first

[1] Note that the orthogonality condition is always fulfilled by the basis vectors that are generated by the PCR and PLS approaches that will be presented later. Furthermore, when using `Matlab`, say, for calculating the eigenvectors, they will be automatically normalized; this means that the practical calculations are rather straightforward

$$\begin{tikzcd} X \arrow[r, bend left, "F^1"] & Z \arrow[r, bend left, "F^2"] & Y \end{tikzcd}$$

Figure 5.1: The dependency model $y = f(x)$ refined

projected onto the internal basis, and from there they are further projected onto the output space, the final projection step being based on MLR regression. Note that because all of the mappings are linear, they can always be combined so that, seen from outside, the original "one-level" model structure is still valid: $Y = (XF^1)F^2 = X(F^1F^2) = XF$.

Now there are approximate mappings instead of only one, as in the MLR case. Is it not so that the regression model will become even more sensitive to noise? However, it is not so. It is not the number of mappings, it is the properties of these mappings that matter — and now, as it turns out, the mapping from input to the latent variables and the mapping from latent variables to output can be made well-conditioned.

The overall regression model construction becomes a two-phase process, so that there are the following tasks:

1. Determine the basis θ.
2. Construct the mapping $F^1 = \theta \left(\theta^T \theta\right)^{-1}$.
3. Calculate the "latent variables" $Z = XF^1$.
4. Construct the second-level mapping $F^2 = \left(Z^T Z\right)^{-1} Z^T Y$.
5. Finally, estimate $\hat{Y}_{\text{est}} = X_{\text{est}} F = X_{\text{est}} F^1 F^2$.

Here Z stands for the internal coordinates corresponding to the training data X and Y. In special cases (for example, for orthonormal θ) some of the above steps may be simplified. The remaining problem is to determine the basis θ so that the regression capability would be enhanced.

How the internal structure should be chosen so that some benefits would be reached? When the rank of the basis is the same as the number of degrees of freedom in the data (normally meaning that there are n basis vectors representing the n dimensional data), the data can be exactly reconstructed, or the mapping between data and the transformed representation can be inverted. This means that also the random noise that is present in the samples will always remain there. A good model, however, should only represent the *relevant* things, ignoring something, hopefully implementing this compression of data in a clever way. In concrete terms, this data compression means dimension reduction, so that there are *fewer* basis vectors than what is the dimension of the data, or $N < n$.

Let us study this a bit closer — assume that the dimension of input is n, the dimension of output is m, the dimension of the latent basis is N, and the number of samples is k. The nominal regression model, matrix F mapping input to output contains $n \cdot m$ free parameters; there are $k \cdot m$ constraint equations. This means that on average, there are

$$\frac{k \cdot m}{n \cdot m} = \frac{k}{n} \tag{5.11}$$

constraints for each parameter. The higher this figure is, the better the estimate becomes in statistical sense, random noise having smaller effect. On the other hand, if the latent basis is used in between the input and output, there is first the mapping from input to the latent basis ($n \cdot N$ parameters) and additionally the mapping from the latent basis to the output ($N \cdot m$ parameters). Altogether the average number of constraints for each parameter is

$$\frac{k \cdot m}{n \cdot N + N \cdot m} = \frac{k}{N\left(1 + \frac{n}{m}\right)}. \tag{5.12}$$

Clearly, if $N \ll n$, benefits can be achieved, or the model sensitivity against random noise can be minimized — of course, assuming that these N latent variables can carry all the relevant information.

How an automatic modeling machinery can accomplish such a clever thing of compression, or “abstracting” the data? There are different views of how the relevant phenomena are demonstrated in the data properties. Speaking philosophically, it is the *ontological assumption* that is left to the user: The user has to decide what are the most interesting features carrying most of the information about the system behavior. Concentrating on different aspects and utilizing the statistical properties of the data accordingly results in different regression methods.

5.2 Principal components

The hypothesis that will now be concentrated on is that *data variance carries information.* This is the assumption underlying *Principal Component Analysis (PCA),* also known as *Karhunen–Loeve decomposition,* and the corresponding regression method PCR. In brief, one searches for the directions in the data space where the data variation is maximum, and uses these directions as basis axes for the internal data model. Whereas noise is (assumed to be) purely random, consistent correlations between variables hopefully reveal something about the real system structure.

Assume that θ_i is the maximum variance direction we are searching for. Data points in X can be projected onto this one-dimensional subspace determined by θ_i simply by calculating $Z_i = X\theta_i$; this gives a vector with one scalar number for each of the k measurement samples in X. The (scalar) variance of the projections can be calculated[2] as $E\{z_i^2(k)\} = \frac{1}{k} \cdot Z_i^T Z_i = \frac{1}{k} \cdot \theta_i^T X^T X \theta_i$. Of

[2] Here, again, maximum degrees of freedom existent in the data is assumed; for example, if the centering for the data is carried out using the sample mean, the denominator should be $k-1$. However, this scaling does not affect the final result, the directions of the eigenvectors

course, there can only exist a solution if the growth of the vector θ_i is restricted somehow; the length of this vector can be fixed, so that, for example, there always holds $\theta_i^T\theta_i = 1$. This means that we are facing a constrained optimization problem (see Sec. 1.2.4) with

$$\begin{cases} f(\theta_i) & = & \frac{1}{k}\cdot\theta_i^T X^T X\theta_i, \qquad \text{and} \\ g(\theta_i) & = & 1-\theta_i^T\theta_i. \end{cases} \tag{5.13}$$

Using the the method of Lagrange multipliers, the optimum solution θ_i has to obey

$$\frac{d\,J(\theta_i)}{d\theta_i} = \frac{d}{d\theta_i}\left(f(\theta_i) - \lambda_i\cdot g(\theta_i)\right) = \mathbf{0} \tag{5.14}$$

or

$$2\frac{1}{k}\cdot X^T X\theta_i - 2\lambda_i\theta_i = \mathbf{0}, \tag{5.15}$$

giving

$$\frac{1}{k}X^T X\cdot\theta_i = \lambda_i\cdot\theta_i. \tag{5.16}$$

Now, the variance maximization has become an *eigenvalue problem* with the searched basis vector θ_i being an eigenvector of the matrix $R = \frac{1}{k}\cdot X^T X$. The eigenvectors of the data covariance matrix are called *principal components.*

Because of the eigenproblem structure, if θ_i fulfills the equation (5.16), so does $\alpha\theta_i$, where α is an arbitrary scalar; it will be assumed that the eigenvectors are always normalized to unit length, so that $\theta_i^T\theta_i = 1$.

The solution to the variance maximization problem is also given by some of the eigenvectors — but there are n of them, which one to choose? Look at the second derivative:

$$\frac{d^2\,J(\theta_i)}{d\theta_i^2} = \frac{2}{k}\cdot X^T X - 2\lambda_i\cdot I. \tag{5.17}$$

To reach the maximum of $J(\theta_i)$, there must hold $d^2\,J(\theta_i)/d\theta_i^2 \le \mathbf{0}$, that is, the second derivative matrix (Hessian) must be semi-negative definite: For any vector ξ there must hold

$$\xi^T\cdot\left(\frac{2}{k}\cdot X^T X - 2\lambda_i\cdot I\right)\cdot\xi \le 0. \tag{5.18}$$

For example, one can select ξ as being any of the eigenvectors, $\xi = \theta_j$:

$$\begin{aligned} &\theta_j^T\cdot\left(\tfrac{2}{k}\cdot X^T X - 2\lambda_i\cdot I\right)\cdot\theta_j \\ &\qquad = \tfrac{2}{k}\cdot\theta_j^T\cdot X^T X\cdot\theta_j - 2\lambda_i\cdot\theta_j^T\theta_j \\ &\qquad = 2\lambda_j\cdot\theta_j^T\theta_j - 2\lambda_i\cdot\theta_j^T\theta_j \\ &\qquad = 2\lambda_j - 2\lambda_i \le 0. \end{aligned} \tag{5.19}$$

This always holds regardless of the value of $1 \le j \le n$ only for the eigenvector θ_i corresponding to the largest eigenvalue.

5.2.1 Eigenproblem properties

Let us study closer the eigenvalue problem formulation (5.16).

Symmetricity and non-negativity of eigenvalues

It seems that the matrix $R = \frac{1}{k} \cdot X^T X$ (or the data covariance matrix) determines the properties of the PCA basis vectors, and, indeed, these properties turn out to be very useful. First, it can be noted that R is *symmetric,* because there holds

$$R^T = \left(\frac{1}{k} \cdot X^T X \right)^T = \frac{1}{k} \cdot X^T X = R. \tag{5.20}$$

Next, let us multiply (5.16) from left by the vector θ_i^T (note that, of course, this vector is rank deficient, and only "one-way" implication can be assumed):

$$\frac{1}{k} \cdot \theta_i^T X^T \cdot X \theta_i = \lambda_i \cdot \theta_i^T \theta_i. \tag{5.21}$$

This expression consists essentially of two dot products ($\theta_i^T X^T \cdot X \theta_i$ on the left, and $\theta_i^T \cdot \theta_i$ on the right) that can be interpreted as squares of vector lengths. Because these quantities must be real and non-negative, and because k is positive integer, it is clear that the eigenvalue λ_i is always *real* and *non-negative.*

Orthogonality of eigenvectors

Let us again multiply (5.16) from left; this time by *another* eigenvector θ_j^T:

$$\theta_j^T R \theta_i = \lambda_i \cdot \theta_j^T \theta_i. \tag{5.22}$$

Noticing that because R is symmetric (or $R = R^T$), there must hold $\theta_j^T R = (R^T \theta_j)^T = (R \theta_j)^T = \lambda_j \theta_j^T$, so that we have an equation

$$\lambda_j \cdot \theta_j^T \theta_i = \lambda_i \cdot \theta_j^T \theta_i, \tag{5.23}$$

or

$$(\lambda_i - \lambda_j) \cdot \theta_j^T \theta_i = 0. \tag{5.24}$$

For $\lambda_i \neq \lambda_j$ this can only hold if $\theta_j^T \theta_i = 0$. This means that for a symmetric matrix R, eigenvectors are *orthogonal* (at least if the corresponding eigenvalues are different; for simplicity, this assumption is here made). Further, because of the assumed normalization, the eigenvectors are *orthonormal.*

The above orthogonality property is crucial. Because of orthogonality, the eigenvectors are uncorrelated; that is why, the basis vectors corresponding to the maximum variance directions can, at least in principle, be extracted one at a time without disturbing the analysis in other directions.

5.2.2 Analysis of the PCA model

Let us study the properties of the variables in the new basis. There are n eigenvectors θ_i corresponding to eigenvalues λ_i; from now on, assume that they are ordered in descending order according to their numerical values, so that $\lambda_i \geq \lambda_j$ for $i < j$. This is possible because it was shown that the eigenvalues are real and positive (note that the `eig` function in `Matlab` does *not* implement this ordering automatically). When the eigenvectors and eigenvalues are presented in the matrix form

$$\Theta = \left(\begin{array}{c|c|c} \theta_1 & \cdots & \theta_n \end{array} \right) \qquad \text{and} \qquad \Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}, \tag{5.25}$$

where the dimension of Θ and Λ is $n \times n$, the eigenproblem can be expressed compactly as

$$\frac{1}{k} X^T X \cdot \Theta = \Theta \cdot \Lambda. \tag{5.26}$$

It was shown that the vectors constituting Θ are orthonormal; this means that the whole matrix Θ also is, so that $\Theta^T = \Theta^{-1}$. Noticing that $X\Theta = Z$ is the sequence of variables as presented in the new latent basis, one can write

$$\frac{1}{k} \cdot Z^T Z = \frac{1}{k} \cdot \Theta^T X^T X \Theta = \Theta^T \Theta \cdot \Lambda = \Lambda. \tag{5.27}$$

What this means is that the new variables are mutually uncorrelated (because their covariance matrix Λ is diagonal); what is more, the eigenvalues λ_i directly reveal the variances of the new variables. Let us elaborate on this a bit closer.

$$\begin{aligned}
&\mathrm{var}\{z_1\} + \cdots + \mathrm{var}\{z_n\} && \\
&\quad = \lambda_1 + \cdots + \lambda_n && \\
&\quad = \mathrm{tr}\{\Lambda\} && \text{Definition of } \textit{matrix trace} \\
&\quad = \mathrm{tr}\{\tfrac{1}{k} \cdot \Theta^T X^T \cdot X\Theta\} && \\
&\quad = \mathrm{tr}\{\tfrac{1}{k} \cdot X^T X \cdot \Theta\Theta^T\} && \text{(See below)} \\
&\quad = \mathrm{tr}\{\tfrac{1}{k} \cdot X^T X\} && \text{Orthonormality of } \Theta \\
&\quad = \tfrac{1}{k} x_1^2 + \cdots + \tfrac{1}{k} x_n^2 && \\
&\quad = \mathrm{var}\{x_1\} + \cdots + \mathrm{var}\{x_n\}. &&
\end{aligned} \tag{5.28}$$

The *matrix trace* used above returns the sum of the diagonal elements of a square matrix. The change of the multiplication order above is motivated by the trace properties: Note that for all square matrices A and B there must hold

$$\mathrm{tr}\{AB\} = \sum_{i=1}^{n_A} \sum_{j=1}^{n_B} A_{ij} B_{ji} = \sum_{j=1}^{n_B} \sum_{i=1}^{n_A} B_{ji} A_{ij} = \mathrm{tr}\{BA\}. \tag{5.29}$$

The above result (5.28) means that the total variability in x is redistributed in z. It was assumed that variance directly carries information — the information content is then redistributed, too. If the dimension is to be reduced, the optimal approach is to drop out those variables that carry least information: If an $N < n$ dimensional basis is to be used instead of the full n dimensional one, it should be constructed as

$$\theta = \left(\begin{array}{c|c|c} \theta_1 & \cdots & \theta_N \end{array} \right), \tag{5.30}$$

where the vectors θ_1 to θ_N are the directions of the most variation in the data space. If one tries to reconstruct the original vector x using the reduced basis variables, so that $\hat{x} = \theta z$, the error

$$\tilde{x} = x - \hat{x} = x - \sum_{i=1}^{N} z_i \cdot \theta_i = \sum_{i=N+1}^{n} z_i \cdot \theta_i \tag{5.31}$$

has the variance

$$E\{\tilde{x}^T(k)\tilde{x}(k)\} = \sum_{i=N+1}^{n} \lambda_i. \tag{5.32}$$

This reveals that the the eigenvalues of $R = \frac{1}{k} \cdot X^T X$ give a straightforward method for estimating the significance of PCA basis vectors; the amount of data variance that will be neglected when basis vector θ_i is dropped is λ_i.

As an example, study the case of Sec. 4.3 again. The eigenvalues of the data covariance matrix are

$$\left\{ \begin{array}{lcl} \lambda_1 & = & 2 \cdot E\{\xi^2(\kappa)\} + \sigma^2 \\ \lambda_2 & = & \sigma^2, \end{array} \right. \tag{5.33}$$

and the corresponding eigenvectors are

$$\theta_1 = \frac{1}{\sqrt{2}} \cdot \left(\begin{array}{c} 1 \\ 1 \end{array} \right) \quad \text{and} \quad \theta_2 = \frac{1}{\sqrt{2}} \cdot \left(\begin{array}{c} -1 \\ 1 \end{array} \right). \tag{5.34}$$

These basis vectors are shown in Fig. 5.2 (on the right); in this example, the data variance was $E\{\xi^2(k)\} = 1$ and the noise variance was $\sigma^2 = 0.01$. In this case, the ratio between the eigenvalues becomes very large, $\lambda_1/\lambda_2 \approx 200$; the basis vector θ_1 is much more important as compared to θ_2. When a reduced basis with only the vector θ_1 is applied, all deviations from the line $x_2 = x_1$ are assumed to be noise and are neglected in the lower-dimensional basis. The *data collinearity problem is avoided altogether.*

5.2.3 Another view of "information"

In the beginning of the chapter it was claimed that it is *variance maximization* that is the means of reaching good data models. But *why* does this seemingly arbitrary assumption really seem to do a good job?

It must be recognized that the main goal in the data compression is to enhance the *signal-to-noise ratio,* so that the amount of misleading disinformation would be minimized as compared to the valuable real information. And it is here that

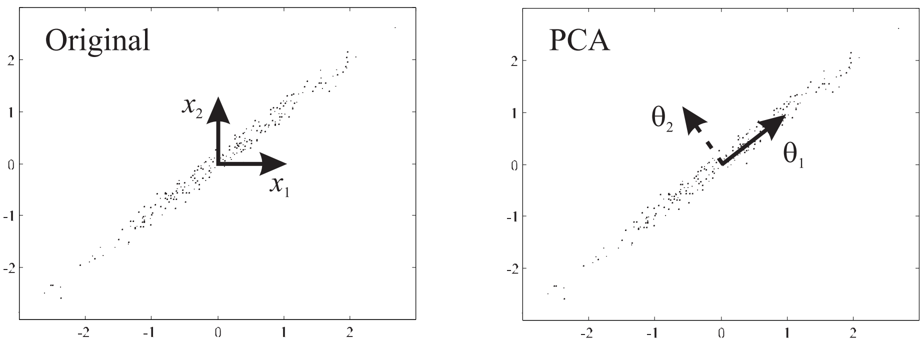

Figure 5.2: Illustration of the "natural" and the PCA bases for the collinear data

the assumptions about "noise ontology" are utilized: The distribution of the noise hopefully differs from that of real information. Typically the underlying basic assumption is that the noise is "more random" than the real signal is; this assumption can have different manifestations:

1. Truly random signals fulfill the assumptions of *central limit theorem,* so that noise distribution is *more Gaussian* than that of real information (this starting point is elaborated on in Chapter 7).

2. If one assumes that noise signals are *uncorrelated* with other signals, the noise is distributed approximately evenly in different directions in the n dimensional space.

The second assumption is utilized in PCA: It is assumed that the same information is visible in various variables, so that the information introduces correlation in the data, whereas noise has no correlations or preferred directions in the data space (see Figs. 5.3 and 5.4). Specially if the data is normalized to unit variance, the *variance pursuit* of PCA changes to *covariance pursuit,* trying to capture the dependencies among variables. The noise variation remaining constant regardless of the direction, the maximum signal-to-noise ratio is reached in the direction where the signal variation is maximum — that is, in the direction of the first principal component. PCR is strongest when MLR is weakest — in large-scale systems with high number of redundant measurements.

Note that PCA gives tools also for further data analysis: For example, if one of the variables varies *alone* (just one variable dominating in the loadings), this variable seemingly does not correlate with other variables — one could consider leaving that variable out from the model altogether (however, see the next section).

5.2.4 Selection of basis vectors

How to determine the dimension of the latent basis? For normalized data $\sum_{i=1}^{n} \lambda_i = n$; a crude approximation is to include only those latent vectors θ_i in the model for which there holds $\lambda_i > 1$ — those directions carry "more

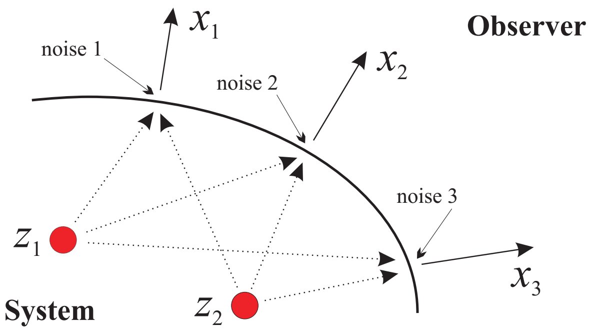

Figure 5.3: Why PCA works: It is assumed that *covariation* reveals some underlying phenomena, whereas noncorrelating variation is measurement noise

that average amount" of the total information (being manifested in the variances). However, the overall behavior of the eigenvalue envelope should be taken into account: That is, plot the eigenvalues in descending order; if there is a significant drop between some of them, this may suggest where to put the model order.

As a rule, it can be argued that the directions of largest eigenvalues are the most important, the dependency relations between variables being concentrated there, whereas the effects of noise are pushed to the later principal components. However, analysis of the components may also reveal some pecularities in the system operation, like outlier data, etc., and the basis selection should not be completely automated. Often the first few eigenvectors represent general dependencies within data, but they may start representing individual disturbances out from the nominal behaviors if these outliers are dominant enough; this all is dependent of the numerical ratios between different phenomena.

If the first principal component dominates excessively, it may be reasonable to check whether the data preprocessing has been successfull: If the data is not mean-centered, it is this mean that dominates in the model rather than the true data variation, specially if the numerical data values are far from origin. The absolute minimum eigenvalue is zero, meaning that the set of measurements is linearly dependent; this can happen also if there are too few measurements, so that $k < n$; note, however, that PCA type data modeling can still be carried out in such case, whereas MLR would collapse. In general, the more there are good-quality samples as compared to the problem dimension, that is, if $k \gg n$, MLR often given good results, whereas the latent basis methods outperform MLR if the number of samples is low (and random variations are visible in data).

If there exist eigenvectors with exactly equal eigenvalues in the covariance matrix, the selection of the eigenvectors is not unique; any linear combination of such eigenvectors also fulfills the eigenvalue equation (5.16). This is specially true for *whitened data,* where the data is preprocessed so that the covariance matrix becomes identity matrix; PCA can find no structure in whitened data (however, see Chapter 7).

It needs to be noted that the PCA results are very dependent of scaling: The principal components can be turned arbitrarily by defining an appropriate or-

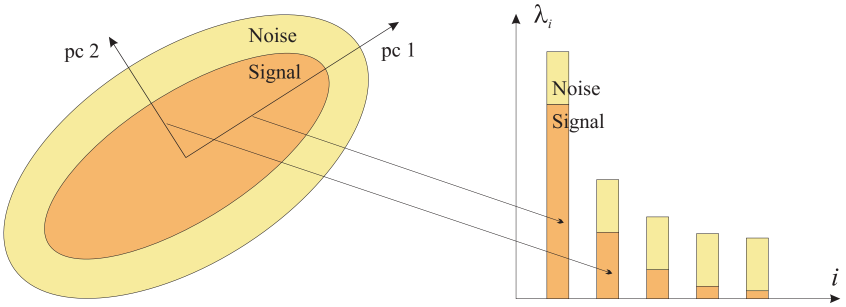

Figure 5.4: Two views of the “directional” information vs. the “undirectional” noise: Five-dimensional data projected onto the first two principal components, on the left, and the corresponding PCA eigenvalues on the right (note that adding a matrix $q \cdot I$, the noise covariance, to the data covariance matrix shifts all eigenvalues up by the amount q). Relatively the most of the noise seems to be concentrated in the directions of lowest overall variation

thogonal transformation matrix D. Assume that $X' = XD$; if there holds $\Lambda = \frac{1}{k} \cdot \Theta^T X^T X \Theta$, then

$$\Lambda = \frac{1}{k} \cdot \Theta^T D \cdot X'^T X' \cdot D^T \Theta, \tag{5.35}$$

so that the new set of eigenvactors is $D^T\Theta$ — directions being freely adjustable.

figure ”InfoNoise”

5.3 Practical aspects

Below, some practical remarks concerning the PCA method are presented. For more theoretical discussions, for the validity of the principal components model, etc., the reader should study, for example, [3].

5.3.1 Regression based on PCA

The PCA approach has been used a long time for data compression and classification tasks. In all applications the basic idea is redundancy elimination — this is the case also in regression applications.

Summarizing, it turns out that the eigenvector corresponding to the largest eigenvalue explains most of the data covariance. The numeric value of the eigenvalue directly determines how much of the data variation is contained in that eigenvector direction. This gives a very concrete way of evaluating the importance of the PCA basis vectors: One simply neglects those basis vectors that have minor visibility in the data. Using this reduced set of vectors as the internal model subspace basis θ_{PCA}, principal component regression (PCR) is directly implemented[3]. Because of the orthogonality of the basis vectors there holds $Z = XF^1 = X\theta_{\text{PCA}}$, and the general modeling procedure (see page 80) reduces into the expression

$$\begin{array}{rcl} F_{\text{PCR}} & = & F^1 F^2 \\ & = & \theta_{\text{PCA}} \left(\theta_{\text{PCA}}^T X^T X \theta_{\text{PCA}}\right)^{-1} \theta_{\text{PCA}}^T X^T Y \\ & = & \theta_{\text{PCA}} \left(k\Lambda_N\right)^{-1} \theta_{\text{PCA}}^T X^T Y. \end{array} \tag{5.36}$$

5.3.2 Other applications

Principal component analysis has routinely been used for data compression tasks, in all kinds of applications where huge amounts of data are being processed. For example, in *neural networks* the input data is often preprocessed in this way to reach manageable adaptation in the network weights — no matter how "outdated" the statistical methods are claimed to be in that community.

PCA has also been applied in more ambitious tasks, hoping that the compression of data would reveal some underlying hidden phenomena. For example, there exist plenty of applications in *fault diagnosis* and *process monitoring.* A rather new solution to these problems is called *multivariate statistical process control (SPC),* where the traditional approach of observing individual variables at a time is extended to analysis of variation structures of multiple variables (see Fig. 5.5).

5.3.3 Analysis tools

The numerical values of the principal component loadings reveal the dependencies (covariances) between different variables, and they also give information about the relevances of different input variables in the regression model. Assuming that $\theta_{i,j}$ is the j'th element in the basis vector i, the contribution of variable z_i when explaining variance in x_j is $\lambda_i \theta_{i,j}^2$, and the overall relevance of this variable is $\hat{\mathrm{E}}\{x_j^2\} = \sum_{i=1}^N \lambda_i \theta_{i,j}^2$, expressing the total amount of variance in x_j that can be reconstructed by the selected latent variables; for normalized x_j this gives a measure for estimating the "value" of that input variable. This kind of analysis is important specially when the model structure is iteratively refined: Non-existent weighting of some of the inputs in all of the latent variables suggests that these inputs could perhaps be excluded from the model altogether.

The PCA model can be analyzed against data in various ways in practice. One can for example calculate the measure for *lack of fit,* the parameter called Q. This is simply the sum of error squares when a data sample is fitted against the reduced basis, and then reconstructed. Because $z(\kappa) = \theta^T x(\kappa)$ and $\hat{x}(\kappa) = \theta z(\kappa)$, there holds $\hat{x}(\kappa) = \theta\theta^T x(\kappa)$, so that the reconstruction error becomes

[3] Even if the basis would not be reduced, the orthogonality of the basis vectors already enhances the numeric properties of the regression model: in a non-orthogonal basis, the different coordinates have to "compete" against each other (heuristically speaking), often resulting in excessive numeric values

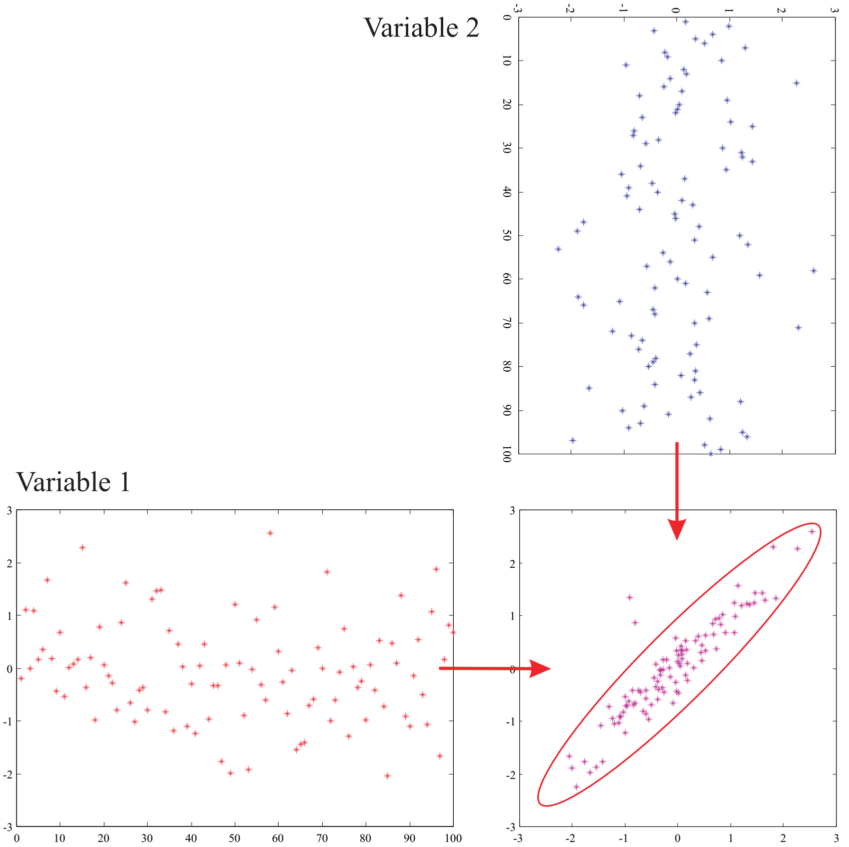

Figure 5.5: Idea of process monitoring using multivariate SPC: It is not always the measurements that are farthest away from the nominal values of the variables that indicate problems in the process

$\tilde{x}(\kappa) = (I_n - \theta\theta^T) \cdot x(\kappa)$. The sum of error squares is then

$$\begin{array}{rcl} Q(\kappa) & = & \tilde{x}^T(\kappa)\tilde{x}(\kappa) \\ & = & x^T(\kappa) \cdot \left(I_n - \theta\theta^T\right)^T \left(I_n - \theta\theta^T\right) \cdot x(\kappa) \\ & = & x^T(\kappa) \cdot \left(I_n - 2\theta\theta^T + \theta\theta^T\theta\theta^T\right) \cdot x(\kappa) \\ & = & x^T(\kappa) \cdot \left(I_n - \theta\theta^T\right) \cdot x(\kappa), \end{array} \tag{5.37}$$

because due to orthonormality of θ there holds $\theta \cdot \theta^T \theta \cdot \theta^T = \theta\theta^T$. The Q statistic indicates how well each sample conforms to the PCA model telling how much of the sample remains unexplained.

Another measure, the *sum of normalized squared scores*, known as *Hotellings T^2 statistic*, reveals how well the data fits the data in another way: It measures the variation in each sample *within* the PCA model. In practice, this is revealed by the scores $z(\kappa)$; the $T^2(\kappa)$ is calculated as a sum of the squared normalized scores. Because the standard deviation of z_i to be normalized is known to be $\sqrt{\lambda_i}$, there holds

$$T^2(\kappa) = z^T(\kappa) \cdot \Lambda_N^{-1} \cdot z(\kappa) = x^T(\kappa) \cdot \theta\Lambda_N^{-1}\theta^T \cdot x(\kappa). \tag{5.38}$$

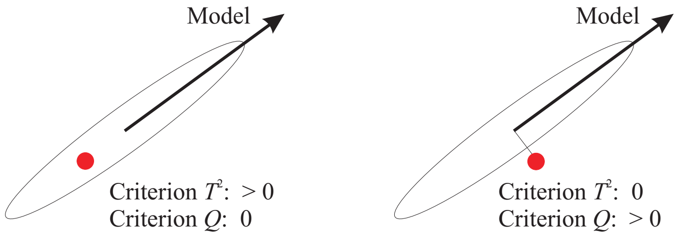

Figure 5.6: The difference between the T^2 and Q criteria: The former data point can be represented within the assumed model, whereas the latter one resides in the subspace that is orthogonal to that model

Roughly speaking, the smaller both of these $Q(\kappa)$ and $T^2(\kappa)$ turn out to be, the better the data fits the model. There are essential differences, though: For example, inflating the basis, or letting N grow, typically increases the value of $T^2(\kappa)$, whereas $Q(\kappa)$ decreases (see Fig. 5.6). Closer analyses could be carried out to find exact statistical confidence intervals for these measures; however, these analyses are skipped here.

5.3.4 Calculating eigenvectors in practice

There exist very efficient methods for calculating eigenvalues and eigenvectors, available, for example, in `Matlab`. However, let us study such a case where the dimension n is very high, and only few of the eigenvectors are needed.

Assuming that the measurement signals are linearly independent, the (unknown) eigenvectors of the covariance matrix span the n dimensional space, that is, any vector ξ can be expressed as a weighted sum of them:

$$\xi = w_1\theta_1 + w_2\theta_2 + \cdots + w_n\theta_n. \tag{5.39}$$

If this vector is multiplied by the covariance matrix, each of the eigenvectors behaves in a characteristic way:

$$R\xi = \lambda_1 \cdot w_1\theta_1 + \lambda_2 \cdot w_2\theta_2 + \cdots + \lambda_n \cdot w_n\theta_n. \tag{5.40}$$

Further, if this is repeated k times:

$$R^k\xi = \lambda_1^k \cdot w_1\theta_1 + \lambda_2^k \cdot w_2\theta_2 + \cdots + \lambda_n^k \cdot w_n\theta_n. \tag{5.41}$$

If some of the eigenvalues is bigger than the others, say, λ_1, finally it starts dominating, no matter what was the original vector ξ; that is, the normalized result equals the most significant principal component θ:

$$\lim_{k\to\infty} \left\{ \frac{R^k\xi}{\|R^k\xi\|} \right\} = \theta_1. \tag{5.42}$$

Assuming that the eigenvalues are distinct, this *power method* generally converges towards the eigenvector θ_1 corresponding to the highest eigenvalue λ_1 —

but only if $w_1 \neq 0$. Starting from a random initial vector ξ this typically holds. However, one can explicitly eliminate θ_1 from ξ, so that

$$\xi' = \xi - \theta_1^T \xi \cdot \theta_1. \tag{5.43}$$

Now there holds

$$\theta_1^T \xi' = \theta_1^T \xi - \theta_1^T \xi \cdot \theta_1^T \theta_1 = 0, \tag{5.44}$$

meaning that θ_1 does not contribute in ξ', and necessarily $w_i = 0$. If the power method is applied starting from this ξ' as the initial guess, the iteration converges towards the eigenvector direction corresponding to the *next highest* eigenvalue λ_2. Further, after the second principal component θ_2 is found, the procedure can be continued starting from ξ'' were both θ_1 and θ_2 are eliminated, resulting in the third eigenvector, etc. If only the most significant eigenvectors are needed, and if the dimension n is high, the power method offers a useful way to iteratively find them in practice (in still more complex cases, where the matrix R itself would be too large, other methods may be needed; see Sec. 8.3.1).

Of course, numerical errors cumulate, but the elimination of the contribution of the prior eigenvectors (5.43) can be repeated every now and then. The elimination of basis vectors can be accomplished also by applying so called *deflation methods* for manipulating the matrix R explicitly.

5.4 New problems

The PCR approach to avoiding the collinearity problem is, however, not a panacea that would always work. To see this, let us study another simple example.

Again, assume that we can observe two variables x_1 and x_2, so that $x = \begin{pmatrix} x_1 & x_2 \end{pmatrix}^T$. This time, however, these variables are independent; and to simplify the analysis further, assume that no noise is present. This means that the covariance matrix becomes

$$\begin{array}{rcl} \frac{1}{k} \cdot X^T X & = & \frac{1}{k} \cdot \begin{pmatrix} \sum_{\kappa=1}^k x_1^2(\kappa) & \sum_{\kappa=1}^k x_1(\kappa) x_2(\kappa) \\ \sum_{\kappa=1}^k x_1(\kappa) x_2(\kappa) & \sum_{\kappa=1}^k x_2^2(\kappa) \end{pmatrix} \\ & \approx & \begin{pmatrix} E\{x_1^2(\kappa)\} & 0 \\ 0 & E\{x_2^2(\kappa)\} \end{pmatrix}. \end{array} \tag{5.45}$$

The eigenvalues are now trivially $\lambda_1 = E\{x_1^2(\kappa)\}$ and $\lambda_2 = E\{x_2^2(\kappa)\}$, and the eigenvectors are $\theta_1 = \begin{pmatrix} 1 & 0 \end{pmatrix}^T$ and $\theta_2 = \begin{pmatrix} 0 & 1 \end{pmatrix}^T$, respectively. If either of the eigenvalues has much smaller numerical value, one is tempted to drop it out (as was done in the previous PCA example). So, assume that θ_2 is left out. What happens if the underlying relationship between x and y can be expressed as $y = f(x_2)$, so that x_1 (or θ_1) is not involved at all? This means that a regression model that uses the reduced PCA basis will *fail completely.*

5.4.1 Experiment: "Associative regression"*

It is evident that one has to take the output into account when constructing the latent variables — so, what if we define

$$v(\kappa) = \left(\frac{x(\kappa)}{y(\kappa)} \right), \tag{5.46}$$

and construct a PCA model for this — then the input and output variables should be equally taken into account in the construction of the latent variables. The corresponding covariance matrix becomes

$$\frac{1}{k} \cdot V^T V = \frac{1}{k} \cdot \left(\begin{array}{c|c} X^T X & X^T Y \\ \hline Y^T X & Y^T Y \end{array} \right), \tag{5.47}$$

so that the eigenproblem can be written as

$$\frac{1}{k} \cdot \left(\begin{array}{c|c} X^T X & X^T Y \\ \hline Y^T X & Y^T Y \end{array} \right) \cdot \left(\frac{\theta_i}{\phi_i} \right) = \lambda_i \cdot \left(\frac{\theta_i}{\phi_i} \right). \tag{5.48}$$

Here, the eigenvectors are divided in two parts: First, θ_i corresponds to the input variables and ϕ_i to outputs. The selection of the most important eigenvectors proceeds as in standard PCA, resulting in the set of N selected eigenvectors

$$\left(\frac{\theta}{\phi} \right). \tag{5.49}$$

The eigenvectors now constitute the mapping between the x and y variables, and the matrices θ and ϕ can be used for estimating y in an "associative way". During regression, only the input variables are known; these x variables are fitted against the "input basis" determined by θ, giving the projected z variables[4]:

$$Z = X \cdot \theta^T (\theta^T \theta)^{-1}. \tag{5.50}$$

The output mapping is then determined by the "output basis" ϕ: Because the coordinates z are known, the estimate is simply

$$\hat{Y} = Z \cdot \phi. \tag{5.51}$$

Combining these gives the regression model

$$F_{\text{ASS}} = \theta^T (\theta^T \theta)^{-1} \phi. \tag{5.52}$$

This should work, at least if the dimension of input n is much higher than that of output m. The problem of loosely connected input and output variables still does not vanish: The correlated variables dominating in the eigenvectors can be in the same block, that is, they may both be input variables or they may both be output variables. Modeling their mutual dependency exclusively may

[4] Note that, whereas the eigenvectors of the whole system are orthogonal, the truncated vectors in θ are not

ruin the value of the regression model altogether. What one needs is a more structured view of the data; the roles of inputs and outputs need to be kept clear during the analysis, and it is the regression models duty to bind them together. This objective is fulfilled when applying the methods that are presented in the following chapter.

It needs to be noted that when concentrating on specific details, something always remains ignored. Now we have seen two methods (MLR and PCA) that offer the best possible solutions to well-defined compact problems. In what follows, MLR will routinely be used when it is justified, and PCA will be used for data compression tasks, understanding their deficiencies; the problems they may possibly ignore are then solved separately. It is expert knowledge to have such a mental "theoretical toolbox" for attacking different problems using appropriate combinations of basic methods depending on the situation at hand.

Computer exercises

1. Study how the data properties affect the principal component analysis; that is, change the degrees of data freedom and noise level (parameters `dofx` and `sigmax`, respectively):

   ```
   dofx = 5;
   sigmax = 0.5;
   X = dataXY(100,10,NaN,dofx,NaN,sigmax);
   regrPCA(X);
   ```

2. Compare the eigenvectors and eigenvalues of the matrix $R = \frac{1}{k} \cdot X^T X$ when the data preprocessing is done in different ways; that is, create data as

   ```
   DATA = dataClust(3,1,100,50,5);
   ```

   and analyze the results of

   ```
   regrShowClust(X,ones(size(X))); hold on; plot(0,0,'o');
   regrPCA(X)
   ```

   when the following approaches are used:

   ```
   X = DATA;
   X = regrCenter(DATA);
   X = regrScale(DATA);
   X = regrScale(regrCenter(DATA));
   X = regrCenter(regrScale(DATA));
   X = regrWhiten(DATA);
   X = regrWhiten(regrCenter(DATA));
   ```

   Explain the qualitative differences in the eigenvalue distributions. Which of the alternatives is recommended for PCR modeling?