

# Numerical Methods for Determining Principal Component Analysis Abstract Factors

---

Béchu, S., Richard-Plouet, M., Fernandez, V., Walton, J., and Fairley, N. (2016) Developments in numerical treatments for large data sets of XPS images. *Surf. Interface Anal.*, doi: 10.1002/sia.5970.

Two complementary methods for computing abstract factors are discussed. These two methods have a common theme of generating eigenvectors and abstract factors one at a time, thus allowing a specific number of abstract factors to be computed rather than being limited to computing all abstract factors for a given problem. Determination of only a limited number of abstract factors represents a significant reduction in the time required to separate signal from noise in a set of data vectors.

## Nonlinear Iterative Partial Least Squares Method

### Introduction

Nonlinear Iterative Partial Least Squares (NIPALS) is a technique for computing eigenvectors and eigenvalues of matrices. Eigenvector computation is the basis for many numerical procedures, not least for providing understandable solutions for ill conditioned systems of equations.

One means of producing an ill-conditioned set of equations is attempting a linear least squares approximation but selecting basis vectors which are only marginally different in direction cosines. While theoretically a solution is possible, finite precision and floating point arithmetic can result in a failure to generate the theoretical inverse required for the solution. A means of obtaining a numerical solution to such problems is to identify the source of the numerical problem and remove the problematic element of the computation process in the hopes of yielding that which is still useful from the problem as initially formulated. One might argue it would be best to reformulate the least squares problem as initially defined to allow an inverse to be obtained, but understanding what went wrong numerically is useful in reformulating the least squares problem so a solution is possible. Eigenanalysis is therefore more than a means of obtaining an answer but also assists in understanding the problem at hand.

The objective of the following sections is to support an understanding for the general statements made in this introductory section. The least squares discussion in the following section provides details and examples which develop ideas aimed at enhancing an understanding for these general points.

### Linear Least Squares Approximation

Given a set of linearly independent functions  $\{f_1(x), f_2(x), f_3(x), \dots, f_m(x)\}$  a function  $y(x)$  can be defined by

$$y(x) = c_1f_1(x) + c_2f_2(x) + c_3f_3(x) + \dots + c_mf_m(x) \quad \dots \quad (1)$$

where  $c_1, c_2, c_3, \dots, c_m$  are constant values.

Example:

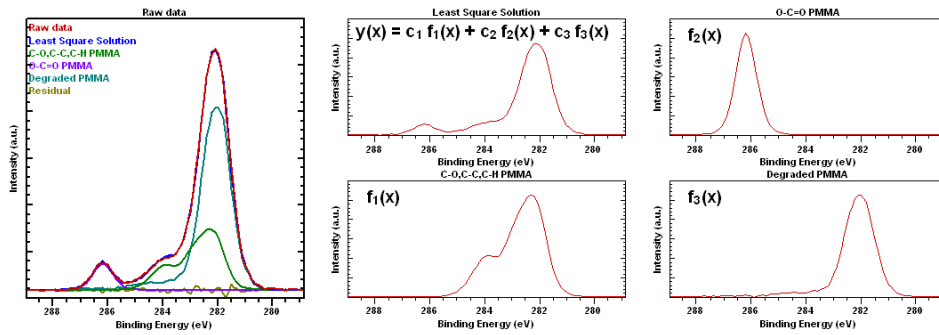


Figure 1: A decomposition for a C 1s spectrum into three components. These components are typically spectra measured from standard materials.

In this well-posed example the functions  $\{f_1(x), f_2(x), f_3(x)\}$  are component curves differing in position, defined by peak maximum, and shape. The weighted sum representing a linear least squares solution of these lineshapes  $y(x)$  approximates a C 1s data envelope.

When using Equation (1) to model a spectrum, the function  $y(x)$  must reproduce the data envelope *as closely as possible*. The concept expressed by the term *closely* has mathematical meaning as follows. The conventional and common measure of closeness is the least squares criterion

$$L(c_1, c_2, c_3, \dots, c_m) = \sum_{i=1}^n (y(x_i) - d_i)^2 \quad \dots \quad (2)$$

is a minimum, where  $\{d_1, d_2, d_3, \dots, d_n\}$  are  $n$  data channels in which signal is collected representing the spectrum of intensity as a function of binding energy (Figure 2).

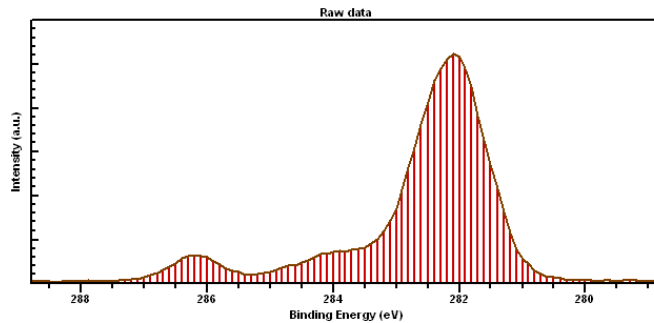


Figure 2: Intensities allocated to energy bins yielding a spectrum which can be logically thought of as a vector with coordinate values corresponding to variation in signal binned by binding energy. The dimension for the vector is equal to the number of data bins.

Minimising the function  $L(c_1, c_2, c_3, \dots, c_m)$  with respect to the parameters  $c_1, c_2, c_3, \dots, c_m$  is achieved by requiring

$$\frac{\partial L}{\partial c_j} = 0 \text{ for all } j = 1, 2, 3, \dots, m$$

Since

$$L(c_1, c_2, c_3, \dots, c_m) = \sum_{i=1}^n (c_1 f_1(x_i) + c_2 f_2(x_i) + c_3 f_3(x_i) + \dots + c_m f_m(x_i) - d_i)^2$$

$$\frac{\partial L}{\partial c_j} = \sum_{i=1}^n 2f_j(x_i)(c_1 f_1(x_i) + c_2 f_2(x_i) + c_3 f_3(x_i) + \dots + c_m f_m(x_i) - d_i) = 0$$

$$\sum_{i=1}^n f_j(x_i)(c_1 f_1(x_i) + c_2 f_2(x_i) + c_3 f_3(x_i) + \dots + c_m f_m(x_i)) = \sum_{i=1}^n f_j(x_i) d_i$$

$$\begin{aligned} c_1 \sum_{i=1}^n f_j(x_i) f_1(x_i) + c_2 \sum_{i=1}^n f_j(x_i) f_2(x_i) + c_3 \sum_{i=1}^n f_j(x_i) f_3(x_i) + \dots + c_m \sum_{i=1}^n f_j(x_i) f_m(x_i) \\ = \sum_{i=1}^n f_j(x_i) d_i \end{aligned}$$

If we use vector notation  $\mathbf{v}_j = (f_j(x_1), f_j(x_2), f_j(x_3), \dots, f_j(x_n))$  and  $\mathbf{d} = (d_1, d_2, d_3, \dots, d_n)$  then, using dot product notation for vector scalar multiplication, the condition for minimising the function  $L(c_1, c_2, c_3, \dots, c_m)$ , namely

$$\frac{\partial L}{\partial c_j} = 0 \text{ for all } j = 1, 2, 3, \dots, m$$

which results in the function  $y(x)$  approximating the data  $\{d_1, d_2, d_3, \dots, d_n\}$  in a least squares sense reduces to a system of linear equations in  $m$  unknowns

$$\begin{aligned} c_1 \mathbf{v}_1 \cdot \mathbf{v}_1 + c_2 \mathbf{v}_1 \cdot \mathbf{v}_2 + c_3 \mathbf{v}_1 \cdot \mathbf{v}_3 + \dots + c_m \mathbf{v}_1 \cdot \mathbf{v}_m &= \mathbf{v}_1 \cdot \mathbf{d} \\ c_1 \mathbf{v}_2 \cdot \mathbf{v}_1 + c_2 \mathbf{v}_2 \cdot \mathbf{v}_2 + c_3 \mathbf{v}_2 \cdot \mathbf{v}_3 + \dots + c_m \mathbf{v}_2 \cdot \mathbf{v}_m &= \mathbf{v}_2 \cdot \mathbf{d} \\ c_1 \mathbf{v}_3 \cdot \mathbf{v}_1 + c_2 \mathbf{v}_3 \cdot \mathbf{v}_2 + c_3 \mathbf{v}_3 \cdot \mathbf{v}_3 + \dots + c_m \mathbf{v}_3 \cdot \mathbf{v}_m &= \mathbf{v}_3 \cdot \mathbf{d} \quad \dots \quad (3) \\ &\vdots \\ c_1 \mathbf{v}_m \cdot \mathbf{v}_1 + c_2 \mathbf{v}_m \cdot \mathbf{v}_2 + c_3 \mathbf{v}_m \cdot \mathbf{v}_3 + \dots + c_m \mathbf{v}_m \cdot \mathbf{v}_m &= \mathbf{v}_m \cdot \mathbf{d} \end{aligned}$$

If a matrix  $\mathbf{A}$  is defined in terms of the vectors  $\mathbf{v}_j$

$$\mathbf{A} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \dots, \mathbf{v}_m]$$

Then the set of simultaneous equations written in matrix notation become

$$\mathbf{A}^T \mathbf{A} \mathbf{c} = \mathbf{A}^T \mathbf{d}$$

where  $\mathbf{c} = (c_1, c_2, c_3, \dots, c_m)$ . The least squares problem has a theoretical solution, provided the inverse matrix  $(\mathbf{A}^T \mathbf{A})^{-1}$  exists, in the form

$$\mathbf{c} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{d} \quad \dots \quad (4)$$

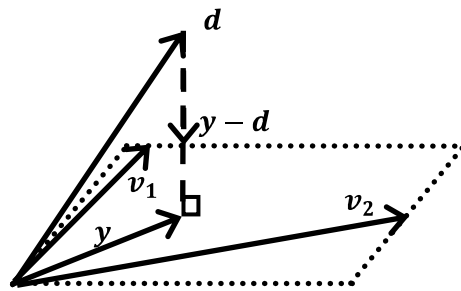
The set of simultaneous equations (3) provide a geometric interpretation for the linear least squares solution  $(c_1, c_2, c_3, \dots, c_m)$ . Each equation is of the form

$$c_1 \mathbf{v}_j \cdot \mathbf{v}_1 + c_2 \mathbf{v}_j \cdot \mathbf{v}_2 + c_3 \mathbf{v}_j \cdot \mathbf{v}_3 + \dots + c_m \mathbf{v}_j \cdot \mathbf{v}_m = \mathbf{v}_j \cdot \mathbf{d}$$

or

$$\begin{aligned} \mathbf{v}_j \cdot (c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + c_3 \mathbf{v}_3 + \dots + c_m \mathbf{v}_m) &= \mathbf{v}_j \cdot \mathbf{d} \\ \Rightarrow \mathbf{v}_j \cdot (c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + c_3 \mathbf{v}_3 + \dots + c_m \mathbf{v}_m) - \mathbf{v}_j \cdot \mathbf{d} &= 0 \\ \Rightarrow \mathbf{v}_j \cdot (\mathbf{y} - \mathbf{d}) &= 0 \end{aligned}$$

That is, the least squares solution creates a vector  $\mathbf{y} = (y(x_1), y(x_2), y(x_3), \dots, y(x_n))$  such that the difference between the least squares solution and the data vector is orthogonal to each of the vectors  $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \dots, \mathbf{v}_m\}$ . A least squares problem involving two functions  $\{f_1(x), f_2(x)\}$  use to approximate a data vector can be visualised as a projection of the data vector onto the plane defined by  $\{\mathbf{v}_1, \mathbf{v}_2\}$ .



While the matrix equation (4) offers a theoretical solution in practice the symmetric matrix  $\mathbf{A}^T \mathbf{A}$  may not be well behaved in the sense that the inverse is difficult to determine from a numerical perspective. The problem occurs when the functions selected for the least squares approximation turn out to be almost the same as others in the function set chosen to describe the data. Such a statement is not always as clear cut as it might sound. For example, the two polynomials  $\{x, x^3\}$  may look different, yet if these represent intensities measured at three point equivalent to  $x = -1, x = 0$  and  $x = 1$  the corresponding matrix is  $\mathbf{A} = \begin{pmatrix} -1 & -1 \\ 0 & 0 \\ 1 & 1 \end{pmatrix}$ , that is two identical vectors yielding  $\mathbf{A}^T \mathbf{A} = \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$ . The polynomials  $x$  and  $x^3$ , for the chosen step size and number of steps, turn out to be linearly dependent functions. If as an alternative the polynomials  $\{x, x^2 - \frac{1}{3}\}$  are chosen for the same number of and size of step, the outcome is very different:

$$\mathbf{A} = \begin{pmatrix} -1 & \frac{2}{3} \\ 0 & -\frac{1}{3} \\ 1 & \frac{2}{3} \end{pmatrix} \text{ and } \mathbf{A}^T \mathbf{A} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$$

Since  $\mathbf{A}^T \mathbf{A}$  is a diagonal matrix the solution represented by Equation (4) is very simple as the inverse matrix  $(\mathbf{A}^T \mathbf{A})^{-1} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{pmatrix}$  is obtain by replacing the non-zero diagonal matrix elements by the

reciprocal of diagonal elements in the original matrix. For this particular example, Equation (4) offers a least squares solution:

$$\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 1 \\ \frac{2}{3} & -\frac{1}{3} & \frac{2}{3} \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix}$$

The reason the solution to the Equation (4) is so easily derived is the polynomials  $\{x, x^2 - \frac{1}{3}\}$  are not only linearly independent but are also orthogonal. The key step in the solution relies on the diagonal matrix elements of  $\mathbf{A}^T \mathbf{A}$  being non-zero while the off diagonal matrix elements are zero. A similar solution for the polynomial pair  $\{x, x^3\}$  is not possible for the given step size and number of steps because the matrix  $\mathbf{A}^T \mathbf{A} = \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$  is singular.

From a practical perspective if a singular matrix is obtained for  $\mathbf{A}^T \mathbf{A}$  then the function set contains at least two functions for which the vectors obtained from these functions are simply a constant multiple of one another.

Since the function set are typically experimental measurements (Figure 1), to an empirical scientist the natural solution is to identify the two vectors which are identical up to a multiplication factor and sum these two vectors together to obtain better signal to noise and replace these two dependent data in the least squares approximation by the sum of these two measurements.

Mathematically, the steps taken by an experimentalist are mimicked by the steps taken by following an algorithm known as Singular Valued Decomposition (SVD). While a full treatment will be left to other texts, essentially the eigenvalues and eigenvectors of the matrix  $\mathbf{A}^T \mathbf{A}$  are calculated and used to create a decomposition of the matrix into an equivalent diagonal matrix consisting of the eigenvalues of  $\mathbf{A}^T \mathbf{A}$ . Whenever a zero or near zero eigenvalue occurs, this is the indicator that the original matrix is near singular and the eigenvector associated with the near zero eigenvalue can be omitted from the linear least squares solution. The SVD algorithm therefore allows for a singular matrix  $\mathbf{A}^T \mathbf{A}$  and delivers back an answer based on a reduced matrix which is not singular, where it is hoped the omitted information is not significant to the problem being solved by a linear least squares approximation.

## Extending LLS to PCA

The above description of a linear least squares approximation solved using an eigenanalysis is intended to give a flavour of how to get an answer from a problem where the problem for one reason or another is poorly defined by *accident*. These ideas, however, apply equally well to problems which are poorly defined on *purpose*. These types of problems are ones in which a large set of data consisting of many measurements result in relatively few distinct functional forms and it is desired to count the actual number of these functional forms rather than needing to examine the individual data measurements for similarities.

Principal Component Analysis (PCA) is performed by defining the least squares problem in terms of data measurements

$$\mathbf{A} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \dots, \mathbf{d}_m]$$

where the vectors  $\mathbf{d}_i$  are individual measurements of spectra or images. The eigenanalysis of the covariance matrix  $\mathbf{A}^T \mathbf{A}$  yields eigenvectors and eigenvalues which allow the eigenvectors to be ordered with respect to the size of the eigenvalues. Since a real symmetric matrix can be geometrically interpreted as a quadratic form involving square and product terms, the act of determining the eigenvalues and eigenvectors provided a method for changing the basis vectors for the quadratic form to a basis set in which the quadratic form is expressed in squared terms without any product terms. The eigenvectors define the directions for the principal axes and the square root of the eigenvalues are interpreted as the sizes for the principal axes for the ellipsoid prescribed by the quadric form defined by the original symmetric matrix  $\mathbf{A}^T \mathbf{A}$ .

In applying the same steps to this, somewhat arbitrary choice of data vectors, as is performed by a LLS approximation, the eigenanalysis orders the information in the data vectors to be consistent with a least squares criterion. That is, the eigenvector with largest eigenvalue represents the greatest variation in the original set of data vectors. Each successive eigenvector ordered by eigenvalues, like the LLS solution, provides the next most important contribution to the data set in terms of content, and by virtue of the eigenanalysis of a real symmetric matrix, these eigenvectors corresponding to distinct eigenvalues are mutually orthogonal.

The result of the eigenanalysis of  $\mathbf{A}^T \mathbf{A}$  allows the identification of vectors  $\mathbf{u}_j$  processed from these eigenvectors and eigenvalues which are representation of the original data set in the sense that a linear combination of the form

$$\mathbf{y}_i = c_{1i}\mathbf{u}_1 + c_{2i}\mathbf{u}_2 + c_{3i}\mathbf{u}_3 + \dots + c_{mi}\mathbf{u}_m \quad \dots \quad (4)$$

can be used to reproduce each of the data vectors  $\mathbf{d}_i$ , that is  $\|\mathbf{d}_i - \mathbf{y}_i\| \approx 0$  (subject only to numerical errors). If instead of using all  $m$  vectors and only including those corresponding to eigenvalues of significant size then the reduced sum ( $r < m$ )

$$\bar{\mathbf{y}}_i = c_{1i}\mathbf{u}_1 + c_{2i}\mathbf{u}_2 + c_{3i}\mathbf{u}_3 + \dots + c_{ri}\mathbf{u}_r \quad \dots \quad (5)$$

results in the possibility of excluding terms only representative of noise and therefore the vector  $\bar{\mathbf{y}}_i$  offers a means of improving signal to noise in a data set. Effectively, each data vector  $\mathbf{d}_i$  is replaced by a linear least squares solution based on trial solution (5).

The formulation in terms of vectors  $\mathbf{u}_j$  is obtained by observing if  $\mathbf{p}_i$  is an eigenvector of  $\mathbf{A}^T \mathbf{A}$  with eigenvalue  $w_i$  then

$$\mathbf{A}^T \mathbf{A} \mathbf{p}_i = w_i \mathbf{p}_i$$

If  $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m]$  and  $\mathbf{W} = \begin{bmatrix} w_1 & 0 & 0 & 0 \\ 0 & w_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & w_m \end{bmatrix}$  then

$$\mathbf{A}^T \mathbf{A} \mathbf{P} = \mathbf{P} \mathbf{W} \Rightarrow \mathbf{P}^{-1} \mathbf{A}^T \mathbf{A} \mathbf{P} = \mathbf{W}$$

and since it is assumed  $\mathbf{P}$  is a matrix chosen to be constructed from a set of orthonormal eigenvectors it follows that  $\mathbf{P}^{-1} = \mathbf{P}^T$ , therefore

$$P^{-1}A^TAP = W \Rightarrow P^T A^T AP = W \Rightarrow (AP)^T AP = W$$

Thus the set of vectors  $U = AP = [u_1, u_2, \dots, u_m]$  are constructed such that  $U^T U$  is a diagonal matrix with diagonal element equal to the eigenvalues of the matrix  $A^T A$ . These n-dimensional vectors  $u_j$  are the abstract factors required to calculate a least squares approximation where the original data set  $A = [d_1, d_2, d_3, \dots, d_m]$  are vectors corresponding to the basis functions equivalent to those in Equation (1).

The only remaining problem is that of computing the eigenvectors of  $A^T A$ . If all eigenvectors are required then the problem is computationally challenging, but fortunately for many applications involving large data set not all the eigenvectors are required. That is Equation (5) is of more interest typically than Equation (4). NIPALS is an algorithm designed to compute eigenvectors one at a time in order of the magnitude of the corresponding eigenvalue.

### NIPALS Algorithm

The first mystery about the NIPALS algorithm is surely the name. It might seem odd to use the term Non-linear when clearly the procedure yields a linear least squares outcome. In the case of this particular algorithm the use of Non-linear in NIPALS is referring to the rate of convergence rather than any non-linearity in the mathematics. One of the most remarkable aspects of the iterative procedure is how for most data sets the first eigenvector is obtain with almost no effort. NIPALS is non-linear in the same sense Newton-Raphson method is non-linear when finding a root of a function. The non-linear relates to the number of significant digits in the value of interest achieved following an iterative cycle. Sadly, like Newton-Raphson, the non-linear convergence is best case behaviour. Worst case convergence can be slow. To understand these convergence issues in NIPALS it is worth considering why the steps now describe do yield the required eigenvectors.

In general terms and by way of illustrating a principle, consider a vector  $x$  of dimension  $m$ , where the dimension for  $x$  corresponds to the number of data vectors forming the matrix  $A = [d_1, d_2, d_3, \dots, d_m]$  and where  $d_i$  are data vectors with  $n$  acquisition channels.

If it is assumed a set of eigenvectors exist and are computed for the matrix  $A^T A$

$$A^T A p_i = w_i p_i$$

and all else being equal, the eigen equation yields  $m$  orthonormal eigenvectors. Since these eigenvector for a real symmetric matrix form a basis set it is possible to write

$$x = \sum_{i=1}^m \alpha_i p_i \quad \dots \quad (6)$$

Thus

$$\begin{aligned} A^T A x &= A^T A \left( \sum_{i=1}^m \alpha_i p_i \right) \\ &= \left( \sum_{i=1}^m \alpha_i A^T A p_i \right) = \left( \sum_{i=1}^m \alpha_i w_i p_i \right) \end{aligned}$$

Therefore repeatedly multiplication by  $A^T A$  results in a transformation of  $x$  as follows.

$$(A^T A)^k x = \left( \sum_{i=1}^m \alpha_i w_i^k p_i \right)$$

The eigenvalues are raised to a power by repeatedly multiplying a vector by the matrix  $A^T A$ . If one eigenvalue is larger than all others, the factor will dominate the summation term. Hence the resulting vector due to iterations is the eigenvector corresponding to the largest eigenvalue.

A further consideration derives from Equation (6). Since  $p_i$  belongs to a set of orthonormal eigenvectors  $p_i \cdot p_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$  and therefore the coefficient in Equation (6) are computed as follows  $\alpha_i = x \cdot p_i$ .

If  $x \rightarrow p_1$  then  $\alpha_1 \rightarrow \|x\|$  and  $\alpha_{i \neq 1} \rightarrow 0$ . These relationships provide an alternative perspective for the converging sequence of vectors. In the event the eigenvalues differ only marginally resulting in a slow movement towards the largest of these similar eigenvalues, a good guess for the initial eigenvector is important to obtain convergence within a reasonable number of iterations.

NIPALS includes iterative steps which rely on these types of transformations. However rather than forming a covariance matrix  $A^T A$  the data matrix  $A$  is used to transform vectors in a sequence leading to computation of  $u_1$ , the abstract factor corresponding to the largest eigenvalue of  $A^T A$ .

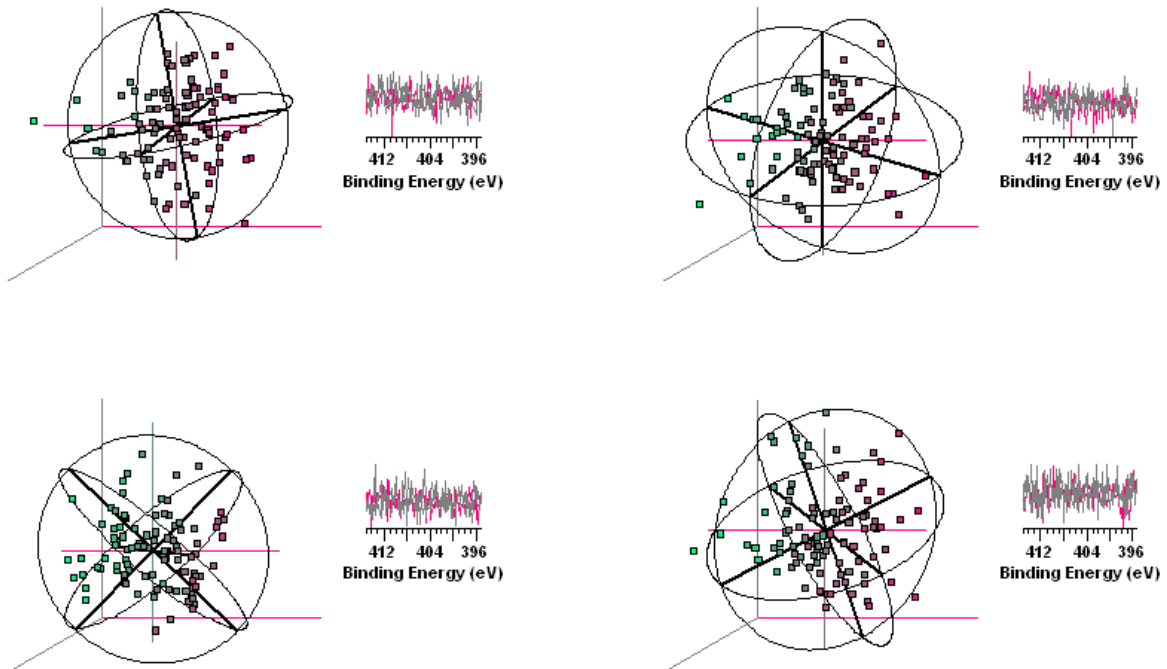


Figure 3: Vectors typical of noise are illustrated using scatter plots with principal axes characterising these four examples of noise distributions. Note how each set of ellipsoids determined for these four distributions result in different orientations for the axes determined from principle component vectors.

The essential iterative steps performed during a NIPALS are as follows.



The input to an iterative step is a matrix  $D_i$ , where initially  $D_1 = A$ . The output from each iterative sequence making use of  $D_i$  is a vector  $u_i$  and a matrix  $D_{i+1}$ . These steps involve selecting an initial vector  $y_0$  of dimension  $n$ . A vector  $x_0$  of dimension  $m$  is computed using the following operations.

$$D_i^T \hat{y}_j = x_j$$

$$D_i \hat{x}_j = y_{j+1}$$

where  $\hat{a}$  is the unit vector corresponding to the vector  $a$ . These steps represent a vector multiplied by a matrix rather than directly constructing a covariance matrix. Separating the action of the covariance matrix into these two intermediate steps is advantageous if convergence to the desired vector is rapid as it is designed to avoid matrix multiplication.

A sequence of vectors  $\{y_j\}$  is constructed which converges to the vector  $u_i$ , corresponding to the largest eigenvalue of  $D_i^T D_i$ . Once the vector  $u_i$  is established the ultimate operation for a single iteration is to deflate  $D_i = [d_1^i, d_2^i, d_3^i, \dots, d_m^i]$  using the computed vector  $u_i$

$$d_j^{i+1} = d_j^i - \frac{d_j^i \cdot u_i}{u_i \cdot u_i} u_i \quad \dots \quad (7)$$

resulting in the next matrix in the NIPALS sequence  $D_{i+1}$ . The new matrix represents a set of vectors all belonging to a subspace of dimension one less than the previous step. The projection operation in Equation (7) is found also in Gram-Schmidt, a procedure for constructing a set of mutually orthogonal vectors. Deflating these  $D_i$  matrices permits the next iteration of the NIPALS procedure to target the next smallest eigenvalue and hence compute the next vector  $u_{i+1}$ .

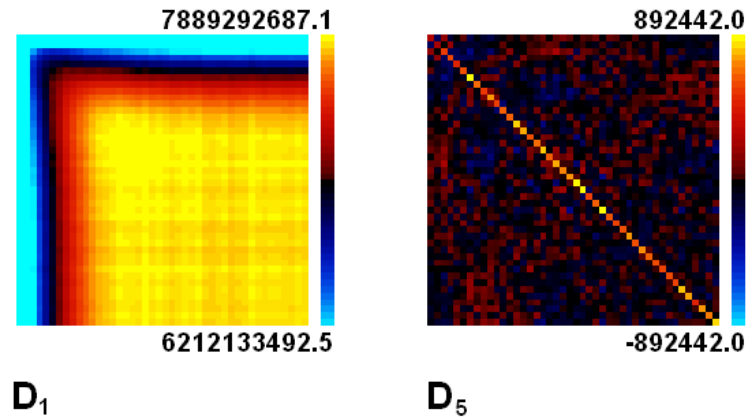


Figure 4: These images represent the covariance matrix  $D_i^T D_i$  constructed for the initial data  $i = 1$  and after four NIPALS iterations  $i = 5$  for a set of Al 2s/Cu 3s spectra. The values in  $D_5^T D_5$  image are small by comparison to  $D_1^T D_1$  and distributed about zero representing vectors consistent with an increasing noise component.

These  $D_i$  matrices progressively become populated by vectors increasingly with the characteristics of noise (Figure 3). The very condition for a rapid convergence is progressively lost following NIPALS steps. As the number of NIPALS iterations increases, data transformations based on rotations of vectors can be used in an attempt to secure an improved estimate for the initial vector for the NIPALS sequence, reducing the number of iterative steps. Such rotations are typically unnecessary

for the initial vectors as NIPALS non-linear behaviour often dominates, but as these  $D_i$  increasingly take on characteristics of noise (Figure 4) with no particular direction of any significance, pre-processing these sets of vectors can reduce the overall number of NIPALS iterations before conditions are met which suggest no further principal components can be determined in acceptable time, and which differ from the expected noise in the original data. The NIPALS iterations terminate typically after a small number of vectors are determined.

If all eigenvectors are required, then other approaches to computing the full set of eigenvectors will be more efficient, however, most applications encountered in CasaXPS do not require all eigenvector and particularly for large data sets NIPALS represents a significant time-saving algorithm, particularly for image processing.

### Iterative SVD Sort Determination of Abstract Factors

A simple but effective iterative method for computing Principal Component Analysis abstract factors is presented. The algorithm is described and illustrated in terms of the covariance matrix which is central to PCA and preparing data in the form of singular value decomposition for a set of data vectors.

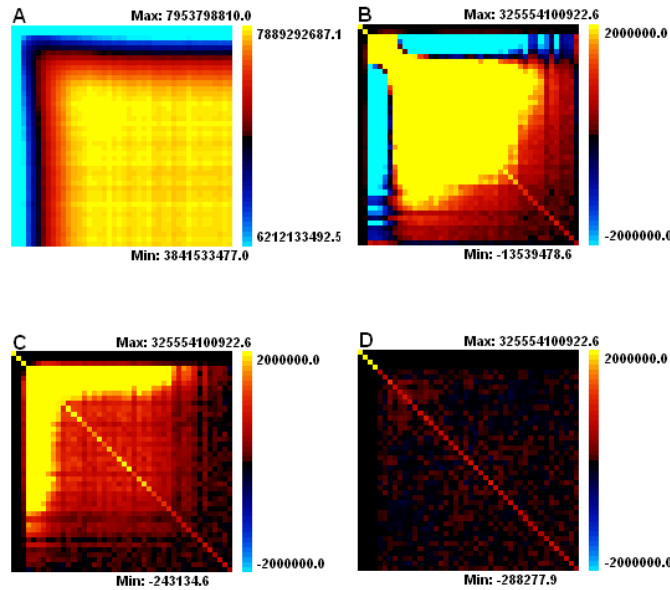


Figure 5: Image representation of covariance matrices formed from vectors following transformation due to computational steps leading to a sequence of principal component abstract factors.

Given a set of data vectors  $\{d_1, d_2, d_3, \dots, d_m\}$ , the standard procedure for expressing these vectors as a corresponding set of abstract factors  $\{u_1, u_2, u_3, \dots, u_m\}$  is in terms of a singular valued decomposition

$$D = UWV^T$$

where

$$d_i \in \mathbb{R}^n, u_i \in \mathbb{R}^n, D = [d_1, d_2, d_3, \dots, d_m] \text{ and } U = [\hat{u}_1, \hat{u}_2, \hat{u}_3, \dots, \hat{u}_m]$$

$W$  is a diagonal matrix with diagonal matrix elements equal to the eigenvalues of the covariance matrix

$$Z = D^T D$$

and  $V$  is the matrix formed from the normalised eigenvectors of  $Z$  ordered with respect to the eigenvalues. The eigenvalues appear ordered in size along the diagonal of  $W$ .

Given a set of vectors derived from spectrum or image data, an impression for a covariance matrix can be presented in image format where dot product between pairs of vectors contribute values to image pixels intensities organised from the rows and columns of the matrix  $Z$ . The image labelled A in Figure 5 is formed from the covariance matrix computed from raw spectral data. The sequence of images in Figure 5 labelled B, C and D represent the state of the set of vectors used to construct image A following calculation of the first three largest eigenvector for the original covariance matrix.

A covariance matrix is symmetrical hence the obvious symmetry about the leading diagonal for these images in Figure 5. These images emphasize how, with each iteration of the algorithm, the off diagonal pixels diminish in absolute value relative to the magnitude of diagonal pixels. Computed eigenvectors result in large values appearing on the diagonal in the top left-hand corner and the orthogonality of these eigenvectors with respect of the entire set of vectors is evident from the low intensity pixels for the rows and columns associated with the computed eigenvectors.

An interesting feature of these images is the algorithm not only identifies an eigenvector but also alters all other vectors to enhance the diagonal terms. If all eigenvectors were computed the image would appear with non-zero pixels along the diagonal only. Each computed eigenvector therefore results in a general sorting of the data into vectors with characteristic typical of an orthogonal set of vectors.

These eigenvectors are computed by sequentially transforming the set of vectors using  $3 \times 3$  matrices calculated from three vectors at a time. The process is an iterative procedure where after each pass through the data set an approximation to the next largest eigenvector is obtained. The steps are defined as follows:

Given a set of vectors  $y_1 \dots y_m$

$$y_k = \begin{pmatrix} a_{1k} \\ a_{2k} \\ a_{3k} \\ \vdots \\ a_{nk} \end{pmatrix}$$

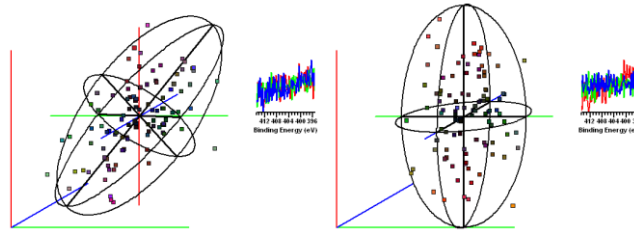
repeat "while current approximation to largest eigenvector has not converged"

loop  $i = m$  down to 3 do

"replace the vectors  $y_1, y_2$  and  $y_i$  by transformed vectors corresponding to the eigenvectors of covariance matrix computed from  $y_1, y_2$  and  $y_i$  in the order of magnitude of the eigenvalues and return the largest eigenvalue."

These remarkably simple steps provide a regime for computing the abstract factor vectors for a data matrix. Each such step places the abstract vector corresponding to the largest eigenvalue of the  $3 \times 3$  covariance matrix in the vector  $y_1$ .

These steps are also suitable for providing an initial vector for input to the NIPALS algorithm, but will converge to the desired vectors representing the abstract factors for a principal component analysis.



**Figure 6:** Three vectors (in this case derive from spectra without obvious peak structure) transform to three vectors such that the scatter plot is aligned with the coordinate axes.

The mechanism for generating an eigenvector is as follows. Given three non-orthogonal vectors  $y_1$ ,  $y_2$  and  $y_i$ , calculating the  $3 \times 3$  covariance matrix and determining the eigenvectors provides a means of defining a matrix which transforms these three original vectors to three vectors which span the same vector subspace as the original three vectors. If the coordinates from these three vectors are used to define the position of points in 3D space (Figure 6), the eigenvectors have the following meaning. The transformation matrix formed from the computed eigenvectors of the  $3 \times 3$  covariance matrix prescribe how to alter the scatter plot of points in 3D space to maximise variation in the scatter points is in the direction of the axes in order of eigenvalue magnitude. Thus each step in the inner loop gathers the information contents into two vectors, the largest of which approximates the desired abstract factor. Each  $3 \times 3$  matrix when calculated from the next vector pushes the least significant information down the set of vectors while at the same time adding to the most significant information at the top of the list. Convergence is accelerated by using  $3 \times 3$  covariance matrix because the middle of the three eigenvectors can alter significantly to accommodate the new vector when added to the top-most two vectors, as stated in the inner loop above, allowing the principal axis with each iteration to steadily progress towards the abstract vector of interest.

A single pass through a set of vectors does not yield the exact abstract factor of interest for the following reason. The first abstract factor must be orthogonal to all vectors in the set. Each new  $3 \times 3$  calculation only ensures the currently used three vectors are mutually orthogonal and necessarily removes the orthogonality from the previously performed transformation. However, each cycle through the inner loop results in the magnitude and therefore influence of the first two vectors increasing compared to the vectors over which the iteration is performed. As a consequence repeating the steps on a set of transformed vectors alters the least significant vectors until the most significant vector is orthogonal to all other transformed vectors.

Orthogonality of the vector to the entire set of vectors is a necessary condition for determining the next abstract factor, it is not however a sufficient condition.

## Mathematical Logic

The covariance matrix is logically embedded within the steps followed by iterative SVD.  $\mathbf{Z}$  by definition is a real symmetric matrix. The Jacobi method for real symmetric matrices solves the eigenproblem by applying Givens transformations to zero two off diagonal elements of  $\mathbf{Z}$  per transformation. Iterative SVD using a  $2 \times 2$  matrix indirectly performs the exact same transformation as a Givens matrix to  $\mathbf{Z}$  by transforming the data vectors. When extending the  $2 \times 2$  approach by using a  $3 \times 3$  matrix to transform data vectors, six rather than two off diagonal elements of  $\mathbf{Z}$  per transformation are zeroed using a matrix of the form

$$\mathbf{C}_p = \begin{bmatrix} a & b & 0 & 0 & 0 & c & 0 & 0 & 0 \\ d & e & 0 & \dots & 0 & f & 0 & \vdots & \vdots \\ 0 & 0 & 1 & 0 & 0 & 0 & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & 0 & \vdots & \vdots & \vdots & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \vdots & \vdots & \vdots \\ g & h & 0 & \dots & 0 & i & 0 & \vdots & 0 \\ 0 & 0 & \vdots & 0 & 0 & 0 & 1 & 0 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \leftarrow \text{row } p \quad \dots \quad (8)$$

$\mathbf{C}_p$  is constructed by the iSVD to have the property

$$\mathbf{C}_p^{-1} = \mathbf{C}_p^T$$

Applying  $\mathbf{C}_p$  to the matrix  $\mathbf{D}$  has the following consequences for the transformed covariance matrix

$$\mathbf{Z}_0 = [\mathbf{D}\mathbf{C}_p]^T [\mathbf{D}\mathbf{C}_p] = \mathbf{C}_p^T \mathbf{D}^T \mathbf{D} \mathbf{C}_p = \mathbf{C}_p^T \mathbf{Z} \mathbf{C}_p$$

specifically

$$\mathbf{C}_p^T \mathbf{Z} \mathbf{C}_p = \begin{pmatrix} x & 0 & x & x & x & 0 & x & x & x \\ 0 & x & x & x & x & 0 & x & x & x \\ x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x \\ 0 & 0 & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x \end{pmatrix} \leftarrow \text{row } p$$

Iterative SVD applies these orthogonal transformations to the data matrix rather than directly to the covariance matrix as follows.

$$\mathbf{D}_0 = \mathbf{D}$$

$$\mathbf{D}_i = \mathbf{D}_{i-1} \mathbf{C}_p$$

Given the sequence of matrices  $\mathbf{D}_i$  and the transformation matrix (8), the following is true

$$\mathbf{D}_i^T \mathbf{D}_i = \mathbf{C}_p^T \mathbf{D}_{i-1}^T \mathbf{D}_{i-1} \mathbf{C}_p$$

Thus applying the orthogonal transformation to the data matrix is logically equivalent to applying the transformation to the corresponding covariance matrix and since Jacobi iterations converge to the eigenvectors and eigenvalues of  $\mathbf{Z}$ , the same is true for iterative SVD.

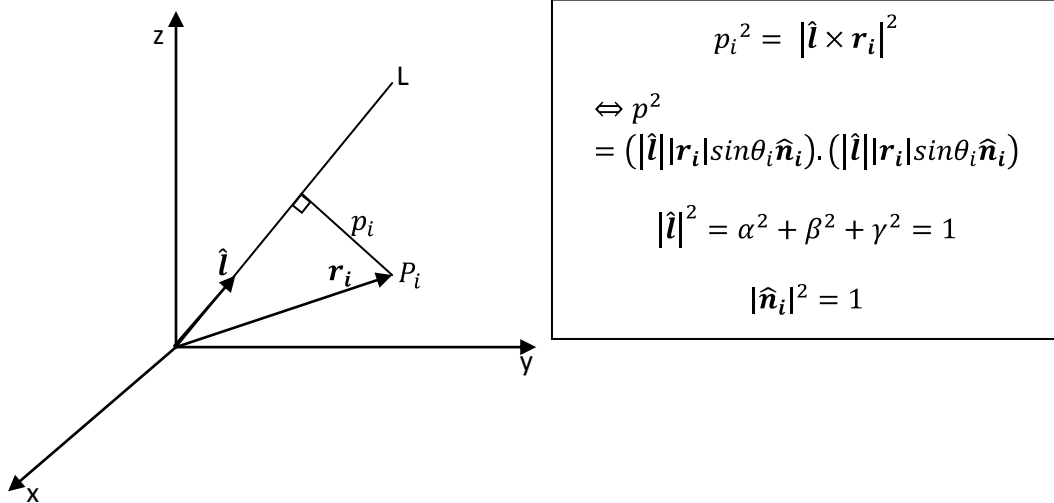
## A Closer Look at PCA and the Covariance Matrix

The use of the covariance matrix  $\mathbf{Z}$  to perform the singular decomposition of the data matrix  $\mathbf{D}$  might seem natural in the context of the design matrix used to perform linear least squares fitting of target vectors to a data vector. However, the use of the covariance matrix formed from the data can be viewed in terms of physics in the following sense. If each component from the three data vectors are viewed as the position in 3D space of a unit mass particle, then the moments of inertia for a set of unit mass particles when rotated with a fixed angular speed attains a minimum kinetic energy for a specific choice of an axis of rotation in 3D space. This problem from classical mechanics is well known and solved using an eigenanalysis for the matrix

$$\begin{bmatrix} A & -H & -G \\ -H & B & -F \\ -G & -F & C \end{bmatrix}$$

where

$$A = \sum_{i=1}^n (y_i^2 + z_i^2), B = \sum_{i=1}^n (x_i^2 + z_i^2), C = \sum_{i=1}^n (y_i^2 + x_i^2), F = \sum_{i=1}^n (y_i z_i), H = \sum_{i=1}^n (x_i y_i) \text{ and } G = \sum_{i=1}^n (x_i z_i)$$



The formulation for this matrix is based on minimizing the moments of inertia  $I$  about a arbitrary line  $L$  passing through the origin for a particle  $P_i$  with unit mass and position vector  $\mathbf{r}_i = (x_i, y_i, z_i)$ . If the perpendicular distance  $p$  from the point  $P$  to the line  $L$  with direction cosines  $\hat{\mathbf{l}} = (\alpha, \beta, \gamma)$ , then

$$I = \sum_{i=1}^n p_i^2$$

$$\text{and since } p_i^2 = |\hat{\mathbf{l}} \times \mathbf{r}_i|^2 = (\beta z_i - \gamma y_i)^2 + (\gamma x_i - \alpha z_i)^2 + (\alpha y_i - \beta x_i)^2$$

$$I = \sum_{i=1}^n [(\beta z_i - \gamma y_i)^2 + (\gamma x_i - \alpha z_i)^2 + (\alpha y_i - \beta x_i)^2]$$

After simplification the moment of inertia for a set of unit mass particles about a line  $L$  reduces to

$$I = A\alpha^2 + B\beta^2 + C\gamma^2 - 2F\beta\gamma - 2G\gamma\alpha - 2H\alpha\beta$$

or written as a quadratic form

$$I = [\alpha \quad \beta \quad \gamma] \begin{bmatrix} A & -H & -G \\ -H & B & -F \\ -G & -F & C \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}$$

Thus the problem of finding the minimum moment of inertia, i.e. minimizing the sum of the square perpendicular distances by choosing the appropriate direction cosines is that of minimising  $I$  subject to the constraint  $\alpha^2 + \beta^2 + \gamma^2 = 1$ , that is, the constraint ensures valid direction cosines are involved in the solution.

Applying the method of Lagrange multiplier to minimising  $I$  subject to  $\alpha^2 + \beta^2 + \gamma^2 = 1$  leads to the requirement to determine eigenvalues and eigenvectors of

$$\mathbf{X} = \begin{bmatrix} A & -H & -G \\ -H & B & -F \\ -G & -F & C \end{bmatrix}$$

So, having observed the requirement to minimise the sum of squares of perpendicular distances to a line yields a minimum moment of inertia for a given distribution of unit mass particles, how does this relate to the covariance matrix  $\mathbf{Z}$ ?

The covariance matrix is formed from the data vectors  $\{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3\}$  as follows:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{d}_1 \cdot \mathbf{d}_1 & \mathbf{d}_1 \cdot \mathbf{d}_2 & \mathbf{d}_1 \cdot \mathbf{d}_3 \\ \mathbf{d}_1 \cdot \mathbf{d}_2 & \mathbf{d}_2 \cdot \mathbf{d}_2 & \mathbf{d}_2 \cdot \mathbf{d}_3 \\ \mathbf{d}_1 \cdot \mathbf{d}_3 & \mathbf{d}_2 \cdot \mathbf{d}_3 & \mathbf{d}_3 \cdot \mathbf{d}_3 \end{bmatrix}$$

The matrix  $\mathbf{X}$  is formed from the data vectors where  $\mathbf{d}_1 = (x_1, x_2, x_3, \dots, x_n)$ ,  $\mathbf{d}_2 = (y_1, y_2, y_3, \dots, y_n)$  and  $\mathbf{d}_3 = (z_1, z_2, z_3, \dots, z_n)$  as follows

$$\mathbf{X} = \begin{bmatrix} \mathbf{d}_2 \cdot \mathbf{d}_2 + \mathbf{d}_3 \cdot \mathbf{d}_3 & -\mathbf{d}_1 \cdot \mathbf{d}_2 & -\mathbf{d}_1 \cdot \mathbf{d}_3 \\ -\mathbf{d}_1 \cdot \mathbf{d}_2 & \mathbf{d}_1 \cdot \mathbf{d}_1 + \mathbf{d}_3 \cdot \mathbf{d}_3 & -\mathbf{d}_2 \cdot \mathbf{d}_3 \\ -\mathbf{d}_1 \cdot \mathbf{d}_3 & -\mathbf{d}_2 \cdot \mathbf{d}_3 & \mathbf{d}_2 \cdot \mathbf{d}_2 + \mathbf{d}_1 \cdot \mathbf{d}_1 \end{bmatrix}$$

Therefore

$$\mathbf{Z} + \mathbf{X} = \begin{bmatrix} \mathbf{d}_1 \cdot \mathbf{d}_1 + \mathbf{d}_2 \cdot \mathbf{d}_2 + \mathbf{d}_3 \cdot \mathbf{d}_3 & 0 & 0 \\ 0 & \mathbf{d}_1 \cdot \mathbf{d}_1 + \mathbf{d}_2 \cdot \mathbf{d}_2 + \mathbf{d}_3 \cdot \mathbf{d}_3 & 0 \\ 0 & 0 & \mathbf{d}_1 \cdot \mathbf{d}_1 + \mathbf{d}_2 \cdot \mathbf{d}_2 + \mathbf{d}_3 \cdot \mathbf{d}_3 \end{bmatrix}$$

$$\mathbf{X} = -\mathbf{Z} + (\mathbf{d}_1 \cdot \mathbf{d}_1 + \mathbf{d}_2 \cdot \mathbf{d}_2 + \mathbf{d}_3 \cdot \mathbf{d}_3) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

If  $\mathbf{v}$  is an eigenvector of  $\mathbf{Z}$  with corresponding eigenvalue  $\lambda$  then

$$\mathbf{Z}\mathbf{v} = \lambda\mathbf{v}$$

and so

$$\mathbf{X}\mathbf{v} = -\mathbf{Z}\mathbf{v} + (\mathbf{d}_1 \cdot \mathbf{d}_1 + \mathbf{d}_2 \cdot \mathbf{d}_2 + \mathbf{d}_3 \cdot \mathbf{d}_3) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{v} = (-\lambda + \mathbf{d}_1 \cdot \mathbf{d}_1 + \mathbf{d}_2 \cdot \mathbf{d}_2 + \mathbf{d}_3 \cdot \mathbf{d}_3)\mathbf{v}$$

Therefore  $\mathbf{v}$  is also an eigenvector of  $\mathbf{X}$  with eigenvalue  $\omega = -\lambda + \mathbf{d}_1 \cdot \mathbf{d}_1 + \mathbf{d}_2 \cdot \mathbf{d}_2 + \mathbf{d}_3 \cdot \mathbf{d}_3$ .

Thus, determining the eigenvectors for  $\mathbf{Z}$  also solves the problem of determining the direction for a line by minimising the sum of the squares of the distances to that line. The least squares problem solved by working with the eigenvectors of the covariance matrix  $\mathbf{Z}$  is now clear.

The eigenvalues  $\omega$  for the matrix  $\mathbf{X}$  are the moments of inertia about the axes of rotation defined by the eigenvectors for which the system is in a possible maxima or minima. The smallest eigenvalue for  $\mathbf{X}$  is the minimum moment of inertia for the given system of unit mass particles. The relationship between the eigenvalues of  $\mathbf{X}$  and the eigenvalues of  $\mathbf{Z}$  is essentially an adjustment of the absolute values for the eigenvalues for  $\mathbf{X}$  relative to a new reference value, namely, half the sum of the extreme moments of inertia for three particles positioned on the three coordinate axes at distances from the origin equal to the magnitude for the three data vectors  $\{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3\}$ , respectively. Specially

$$\omega + \lambda = \mathbf{d}_1 \cdot \mathbf{d}_1 + \mathbf{d}_2 \cdot \mathbf{d}_2 + \mathbf{d}_3 \cdot \mathbf{d}_3$$

Further, since the eigenvalues of  $\mathbf{Z}$  are the roots of the characteristic polynomial

$$\det \begin{bmatrix} (\mathbf{d}_1 \cdot \mathbf{d}_1 - \lambda) & \mathbf{d}_1 \cdot \mathbf{d}_2 & \mathbf{d}_1 \cdot \mathbf{d}_3 \\ \mathbf{d}_1 \cdot \mathbf{d}_2 & (\mathbf{d}_2 \cdot \mathbf{d}_2 - \lambda) & \mathbf{d}_2 \cdot \mathbf{d}_3 \\ \mathbf{d}_1 \cdot \mathbf{d}_3 & \mathbf{d}_2 \cdot \mathbf{d}_3 & (\mathbf{d}_3 \cdot \mathbf{d}_3 - \lambda) \end{bmatrix} = 0$$

The sum of the roots for the cubic polynomial is equal to minus the coefficients of  $\lambda^2$ , namely,  $\mathbf{d}_1 \cdot \mathbf{d}_1 + \mathbf{d}_2 \cdot \mathbf{d}_2 + \mathbf{d}_3 \cdot \mathbf{d}_3$ , therefore the three eigenvalues of  $\mathbf{Z}$  ( $\lambda_1, \lambda_2$  and  $\lambda_3$ ) are such that

$$\mathbf{d}_1 \cdot \mathbf{d}_1 + \mathbf{d}_2 \cdot \mathbf{d}_2 + \mathbf{d}_3 \cdot \mathbf{d}_3 = \lambda_1 + \lambda_2 + \lambda_3$$

Thus the three eigenvalues of  $\mathbf{X}$  are  $\omega_1 = \lambda_2 + \lambda_3$ ,  $\omega_2 = \lambda_1 + \lambda_3$  and  $\omega_3 = \lambda_1 + \lambda_2$ .

A point worth highlighting is the eigenvalues of  $\mathbf{X}$  are physically significant and represent the values for the moments of inertia  $I$  about the principal axes. The principal axes are in the directions defined by the eigenvectors of  $\mathbf{X}$ , and since if  $\mathbf{P}$  is a matrix formed from the eigenvectors of  $\mathbf{X}$  and  $\mathbf{\Omega}$  is a diagonal matrix with non-zero elements equal to the, assumed distinct, corresponding eigenvalues of  $\mathbf{X}$ , namely

$$\mathbf{\Omega} = \begin{bmatrix} \omega_1 & 0 & 0 \\ 0 & \omega_2 & 0 \\ 0 & 0 & \omega_3 \end{bmatrix}$$

then

$$\mathbf{X} = \mathbf{P}\mathbf{\Omega}\mathbf{P}^T$$

Thus, the moments of inertia  $I$  about an axis with direction cosines  $[\alpha \quad \beta \quad \gamma]$  is given by



$$I = [\alpha \quad \beta \quad \gamma] \mathbf{X} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}$$

$$\Rightarrow I = [\alpha \quad \beta \quad \gamma] \mathbf{P} \mathbf{\Omega} \mathbf{P}^T \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}$$

If  $[\alpha_1 \quad \beta_1 \quad \gamma_1]$  is the eigenvector corresponding to eigenvalue  $\omega_1$  then since the eigenvectors are constructed to be orthonormal, that is,  $\alpha_i \alpha_j + \beta_i \beta_j + \gamma_i \gamma_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$

$$\mathbf{P}^T \begin{bmatrix} \alpha_1 \\ \beta_1 \\ \gamma_1 \end{bmatrix} = \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \beta_1 \\ \gamma_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \text{ and similarly } [\alpha_1 \quad \beta_1 \quad \gamma_1] \mathbf{P} = [1 \quad 0 \quad 0]$$

Therefore the moments of inertia about the principal axis with direction  $[\alpha_1 \quad \beta_1 \quad \gamma_1]$  is

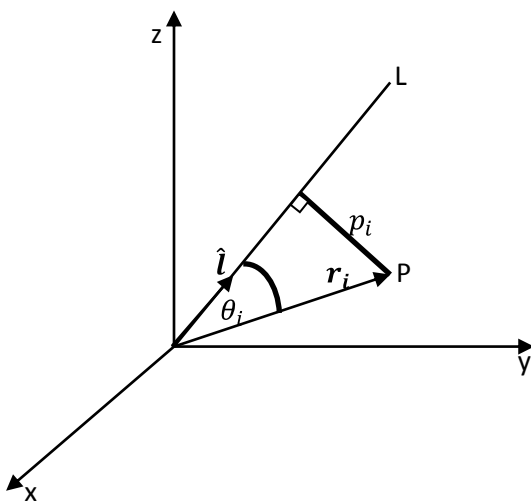
$$I = [\alpha_1 \quad \beta_1 \quad \gamma_1] \mathbf{P} \begin{bmatrix} \omega_1 & 0 & 0 \\ 0 & \omega_2 & 0 \\ 0 & 0 & \omega_3 \end{bmatrix} \mathbf{P}^T \begin{bmatrix} \alpha_1 \\ \beta_1 \\ \gamma_1 \end{bmatrix} = [1 \quad 0 \quad 0] \begin{bmatrix} \omega_1 & 0 & 0 \\ 0 & \omega_2 & 0 \\ 0 & 0 & \omega_3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \omega_1$$

### Covariance Matrix and Least Squares Optimisation

While moments of inertia provide a physical interpretation for PCA, the common practice of working with the covariance matrix  $\mathbf{Z}$  can be understood by following through the logic of minimising the sum of squares of the perpendicular distances from each point in a scatter plot to the principal axis line. If Pythagoras is used to determine the distance from a point in the scatter plot to a line through the origin with direction cosines  $\hat{\mathbf{l}} = (\alpha, \beta, \gamma)$ , the minimisation problem can be expressed as follows.

$$I = \sum_{i=1}^n p_i^2$$

subject to the constraint  $\alpha^2 + \beta^2 + \gamma^2 = 1$



This time, applying Pythagoras theorem to determining the distance  $p_i$

$$p_i^2 = |\mathbf{r}_i|^2 - (|\mathbf{r}_i| \cos \theta_i)^2$$

$$\cos \theta_i = \frac{\mathbf{r}_i \cdot \hat{\mathbf{l}}}{|\mathbf{r}_i| |\hat{\mathbf{l}}|}$$

Since

$$|\hat{\mathbf{l}}|^2 = \alpha^2 + \beta^2 + \gamma^2 = 1$$

$$p_i^2 = |\mathbf{r}_i|^2 - (\mathbf{r}_i \cdot \hat{\mathbf{l}})^2$$

Each point in the scatter plot has coordinates  $\mathbf{r}_i = (x_i, y_i, z_i)$ , therefore

$$p_i^2 = x_i^2 + y_i^2 + z_i^2 - (\alpha x_i + \beta y_i + \gamma z_i)^2$$

and

$$I = \sum_{i=1}^n (x_i^2 + y_i^2 + z_i^2 - (\alpha x_i + \beta y_i + \gamma z_i)^2)$$

The covariance matrix is derived by applying the method of Lagrange multipliers to include the constraint  $\alpha^2 + \beta^2 + \gamma^2 = 1$ , namely the optimisation of the parameters  $\alpha, \beta$  and  $\gamma$  for the function

$$\Psi = \lambda(\alpha^2 + \beta^2 + \gamma^2 - 1) + I$$

Differentiating with respect to  $\alpha$  yields

$$\frac{\partial \Psi}{\partial \alpha} = 2\alpha\lambda + \sum_{i=1}^n (-2(\alpha x_i + \beta y_i + \gamma z_i)x_i) = 2\alpha\lambda - 2\alpha \sum_{i=1}^n x_i^2 - 2\beta \sum_{i=1}^n x_i y_i - 2\gamma \sum_{i=1}^n x_i z_i$$

Equating to zero yields

$$\alpha \sum_{i=1}^n x_i^2 + \beta \sum_{i=1}^n x_i y_i + \gamma \sum_{i=1}^n x_i z_i = \alpha\lambda$$

Similarly,  $\frac{\partial \Psi}{\partial \beta} = 0$  and  $\frac{\partial \Psi}{\partial \gamma} = 0$  provides two more equations as follows.

$$\alpha \sum_{i=1}^n x_i y_i + \beta \sum_{i=1}^n y_i^2 + \gamma \sum_{i=1}^n y_i z_i = \beta\lambda$$

$$\alpha \sum_{i=1}^n x_i z_i + \beta \sum_{i=1}^n y_i z_i + \gamma \sum_{i=1}^n z_i^2 = \gamma\lambda$$

Using  $\mathbf{d}_1 \cdot \mathbf{d}_1 = \sum_{i=1}^n x_i^2$ ,  $\mathbf{d}_2 \cdot \mathbf{d}_2 = \sum_{i=1}^n y_i^2$ ,  $\mathbf{d}_3 \cdot \mathbf{d}_3 = \sum_{i=1}^n z_i^2$ ,  $\mathbf{d}_1 \cdot \mathbf{d}_2 = \sum_{i=1}^n x_i y_i$ ,  $\mathbf{d}_1 \cdot \mathbf{d}_3 = \sum_{i=1}^n x_i z_i$  and  $\mathbf{d}_2 \cdot \mathbf{d}_3 = \sum_{i=1}^n y_i z_i$  and expressing the above simultaneous equations in matrix form the eigenvector problem expressed in terms of the original data vectors results as follows.

$$\begin{bmatrix} \mathbf{d}_1 \cdot \mathbf{d}_1 & \mathbf{d}_1 \cdot \mathbf{d}_2 & \mathbf{d}_1 \cdot \mathbf{d}_3 \\ \mathbf{d}_1 \cdot \mathbf{d}_2 & \mathbf{d}_2 \cdot \mathbf{d}_2 & \mathbf{d}_2 \cdot \mathbf{d}_3 \\ \mathbf{d}_1 \cdot \mathbf{d}_3 & \mathbf{d}_2 \cdot \mathbf{d}_3 & \mathbf{d}_3 \cdot \mathbf{d}_3 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \lambda \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}$$

Thus, the covariance matrix  $\mathbf{Z}$  is recovered from the optimisation problem.

Normalising the eigenvectors determined from the covariance matrix and forming a matrix from these normalised eigenvectors provides, by matrix multiplication, a means of transforming the set of scatter points formed from the input vectors  $\mathbf{d}_i$  to a new set of 3D points corresponding to abstract vectors  $\mathbf{u}_i$  for which the relative positions of the scatter points formed from the coordinates of  $\mathbf{u}_i$  are the same as for the scatter points formed from  $\mathbf{d}_i$ . These ideas are illustrated in the Appendix. Further, the relationship between the covariance matrix eigenvector solution and optimisation

based on a least squares criterion constructs these abstract vectors in directions which account for the variation in the data in order of significance. The most significant information is moved to one abstract vector and the least significant information is moved to another. The ability to partition information is the reason iterative SVD sort offers a means of calculating the next most significant abstract factor in a larger set of vectors.

## Conclusion

SVD Sort and NIPALS are complementary in the sense the output from iterative SVD sort can be used as the input to NIPALS with the implied reduction of NIPALS iterations as a consequence of beginning NIPALS with a vector already closely approximating the desired abstract factor. Nevertheless, iterative SVD sort is capable of generating the desired abstract factors without the need to apply NIPALS iterations with the following advantage.

Gram Schmidt allows the transformation of any set of linearly independent vectors to a mutually orthogonal set of vectors. Projection operations are the means by which Gram Schmidt computes orthogonal vectors. Any mutually orthogonal set of vectors used to construct a covariance matrix results in a diagonal covariance matrix, but without the eigenanalysis of the original covariance matrix these Gram Schmidt vectors are not the desired least squares solution.

NIPALS differs slightly from the iterative SVD sort approach in that after an approximation to an abstract factor is obtained the data set is reduced in dimension by applying a projection operation. This projection operation is similar to Gram Schmidt and is used to remove the computed abstract factor from the set of data vectors. The quality of this vector as an approximation to the desired abstract factor is therefore important at each step, since removing a vector which poorly approximates the abstract factor will alter the nature of the reduced data set and hence the eigenvectors subsequently determined. The projection operation ensures the set of vectors determined will be orthogonal and provide off diagonal elements in a covariance matrix for the final set of vectors which are clearly close to zero. Thus the projection operation removes any feedback suggesting the procedure generated vectors different from the desired abstract factors.

Iterative SVD sort on the other hand does not require projection as the means of reducing the dimension within the set of vectors. Rather, the  $m$  vectors are partitioned into a new abstract factor approximation and  $m - 1$  vectors approximately orthogonal to the new abstract factor. The subsequent abstract factor is obtained by processing the set of  $m - 1$  vectors hence the final set of vectors constructed by iterative SVD sort approach does not force orthogonality but extracts orthogonality. As a consequence the final set of vectors can be assessed by constructing a covariance matrix. Off diagonal covariance matrix elements act as indicators the procedure generated a true representation of the principal component abstract factors.

## Appendix

### Graphical Interpretation of Iterative SVD Sort

The following scatter plots are designed to illustrate the transformations to the original data vectors resulting from iterative SVD sorting six data vectors.

These scatter plots in Figures 7 through 13 are selected to highlight the relationship between adjacent vectors in the list of six vectors. Spectra are plotted as scatter diagrams as follows. The top left tile in each figure represents the first two vectors in the list indicated in Figure 7. Since the first abstract factor computed by iterative SVD sort is placed in the first vector in the list, the top left tile follows the abstract factor of interest. The additional three scatter plots present the other five vectors plotted as sequential sets of three vectors.

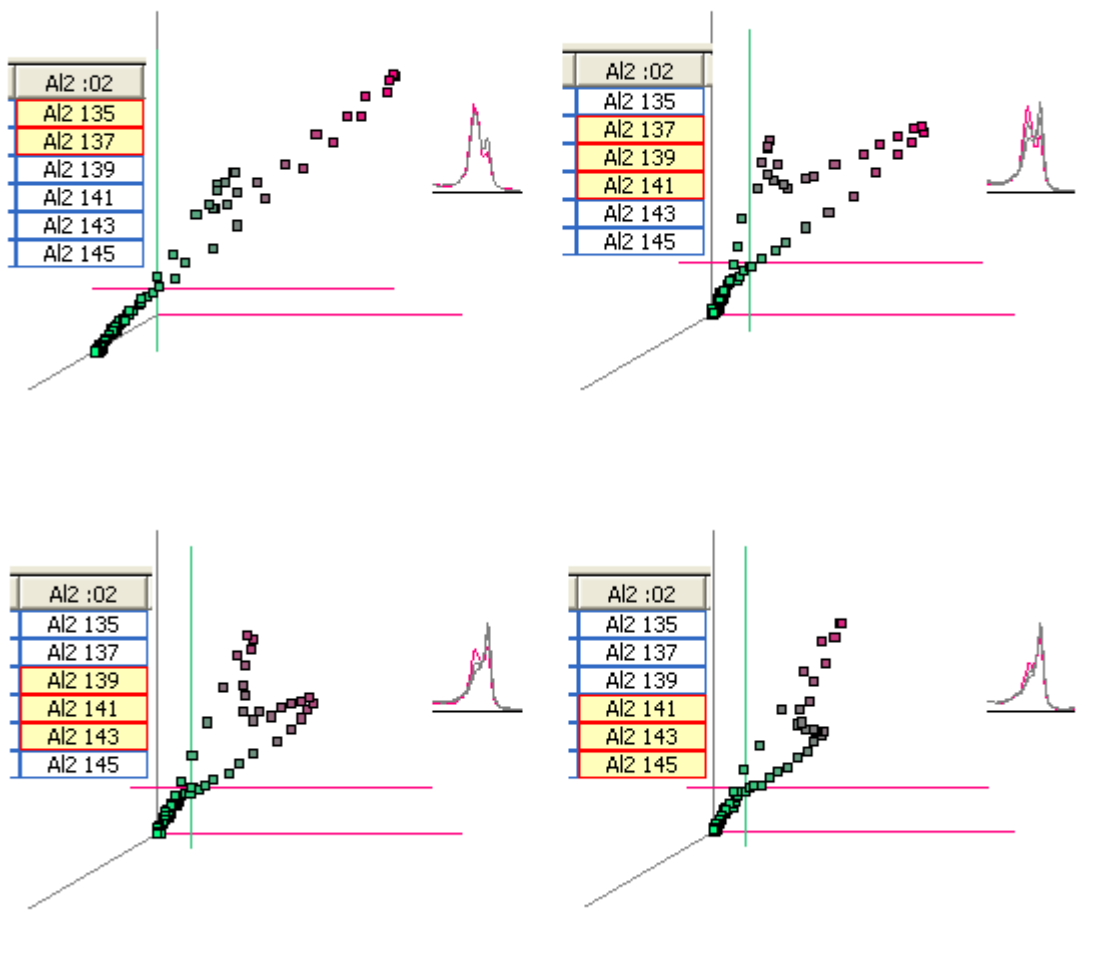


Figure 7: Initial set of data vectors. Note how these sets of vectors plotted as points in 3D space using the coordinate from each vector shown in the display tile initially appear as distributions with clear directions, which by virtue of the evolving set of vectors initially appear to favour the leading diagonal of a cube.

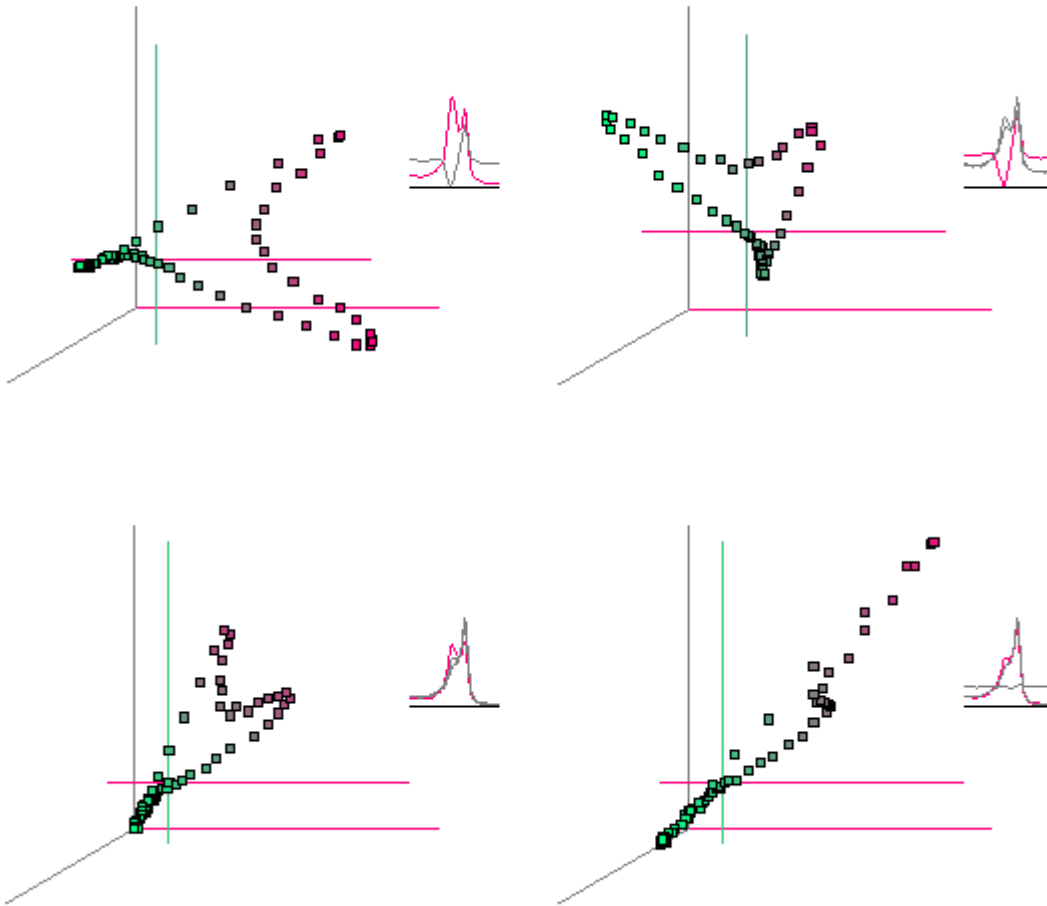


Figure 8: The first step within the first iteration of the SVD sort transforms the first two vectors combined as a set of three vectors with the last vector in the list of vectors. The top left tile traces the relationship between the first two vectors in the list and is therefore altered by the 3x3 eigenanalysis. The two right most tiles both include a vector transformed by this first step.

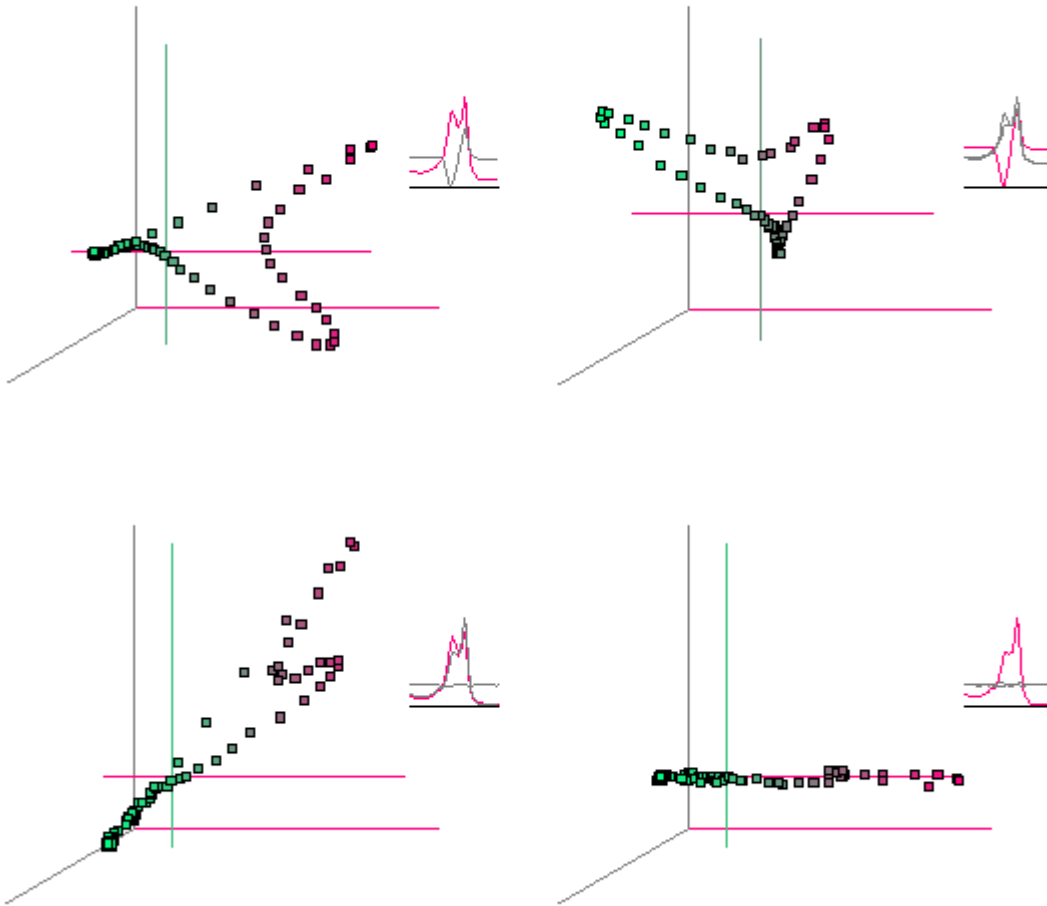


Figure 9: The second step in the first iteration transforms a vector from the three vectors displayed in the bottom-left tile by performing a 3x3 eigenanalysis with the first two vectors. Two vectors are yet to be modified.

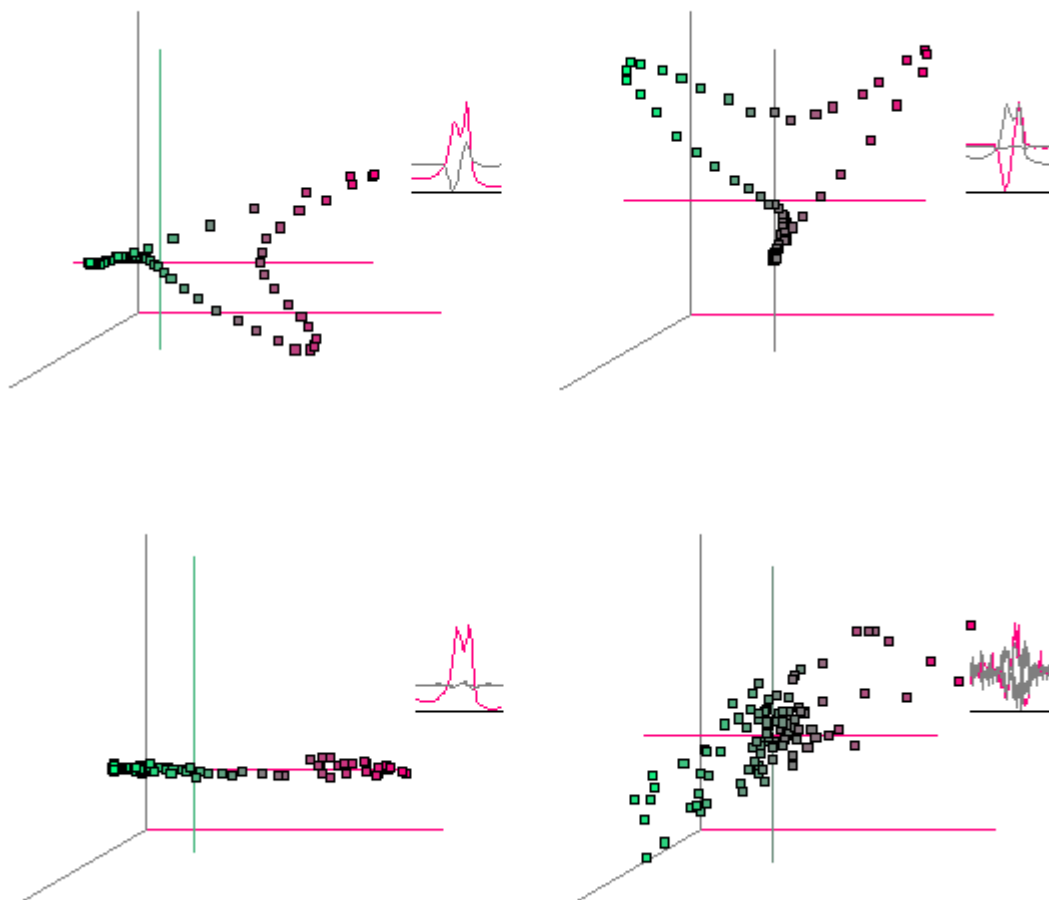


Figure 10: The third step introduces the last but one unmodified vectors into the calculation. Each vector is combined with the first two vectors from the previous step yielding these new vectors which alter the scatter plots.

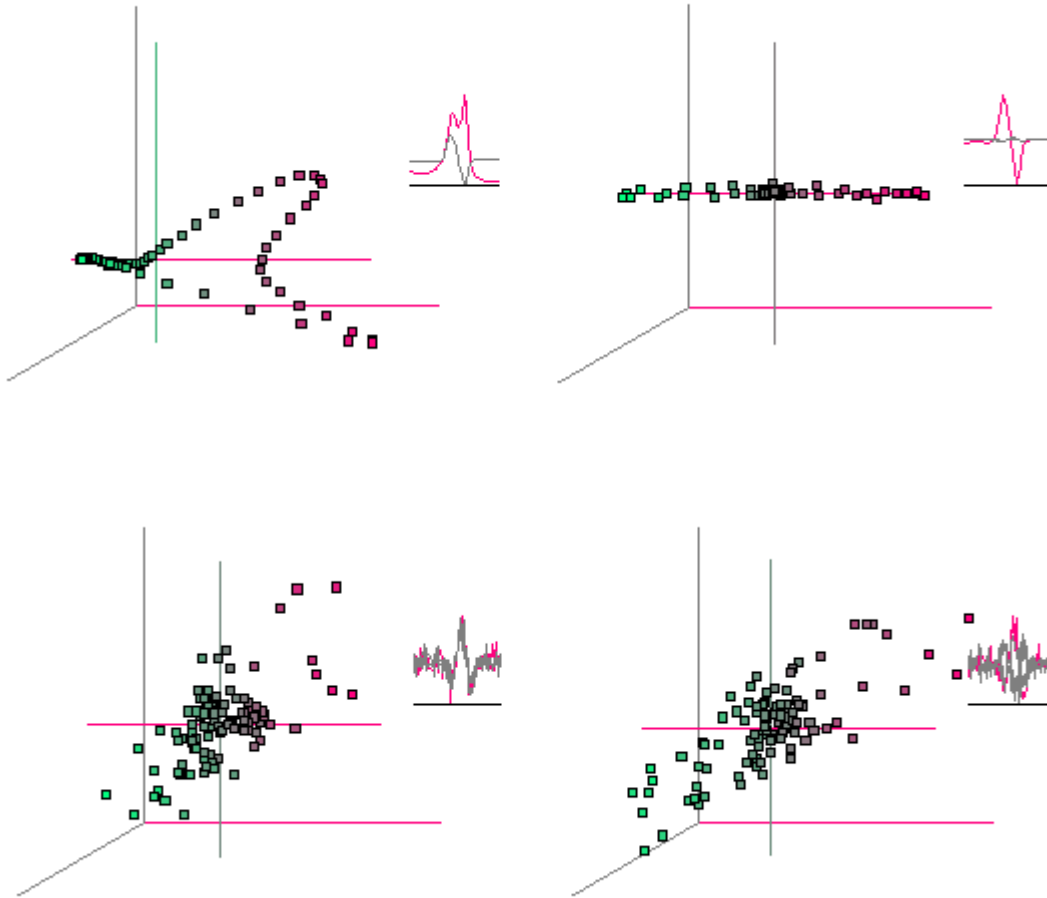


Figure 11: The final unmodified vector is now transformed by computing the eigenvectors for a 3x3 matrix formed from the first two vectors and the yet to be modified vector. These scatter plots represent the state of play after one cycle through the set of six vectors. Each calculation modifies three vectors at a time and following transformation the three vectors represent three mutually orthogonal vectors where the maximum variation is expressed in the first vector in the list. The least variation within the three vectors is moved into the third vector. As a result the first two vectors in the list of six vectors are always orthogonal to each other but only orthogonal to the third vector in the list after a full cycle.



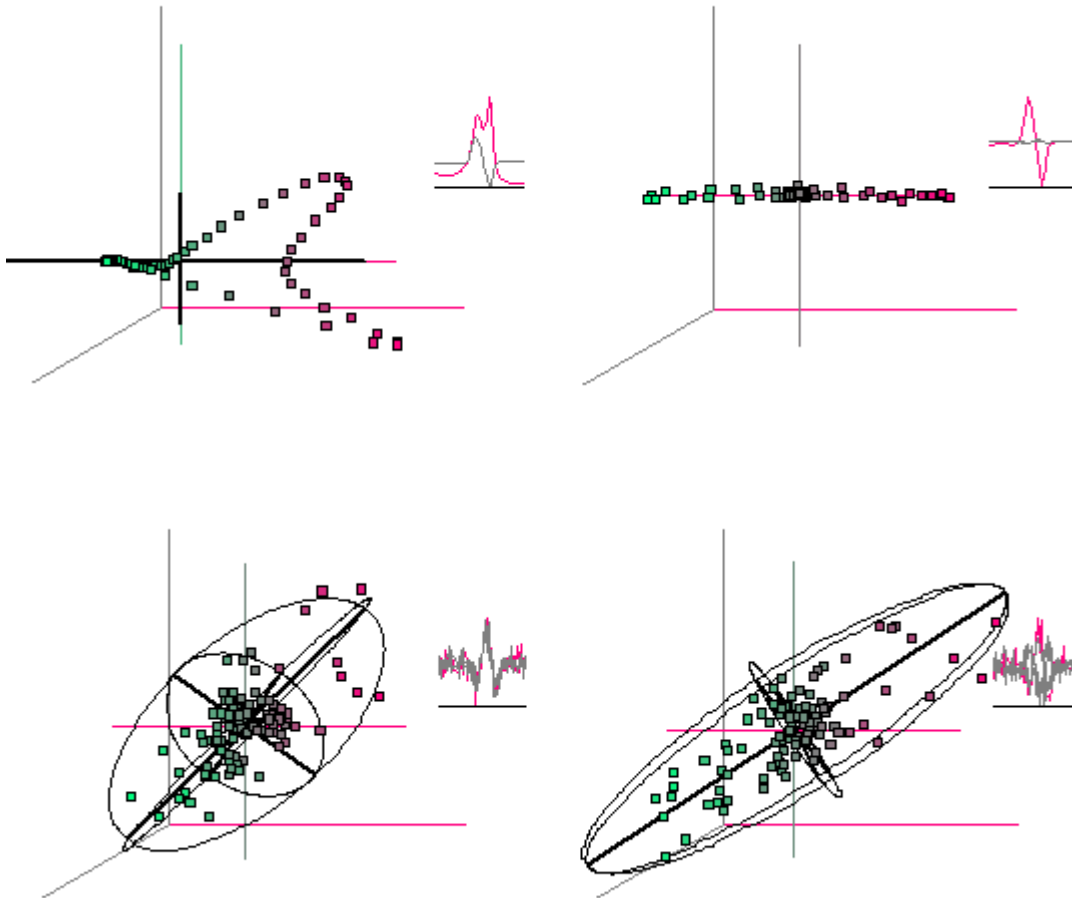


Figure 12: After a number of iterations applied to the list of six vectors the first vector in the list approaches an approximation to the first abstract factor. Each time a new vector is added to the first two vectors and an eigenanalysis is performed the data set are nudged in the direction of vectors orthogonal to the first two vectors. The first vector in the list converges to a vector which is computed to have the maximum variation with respect to each set of three vectors on a triple basis and is orthogonal to all other vectors. The set of vectors not including the first vector are left in an intermediate state.

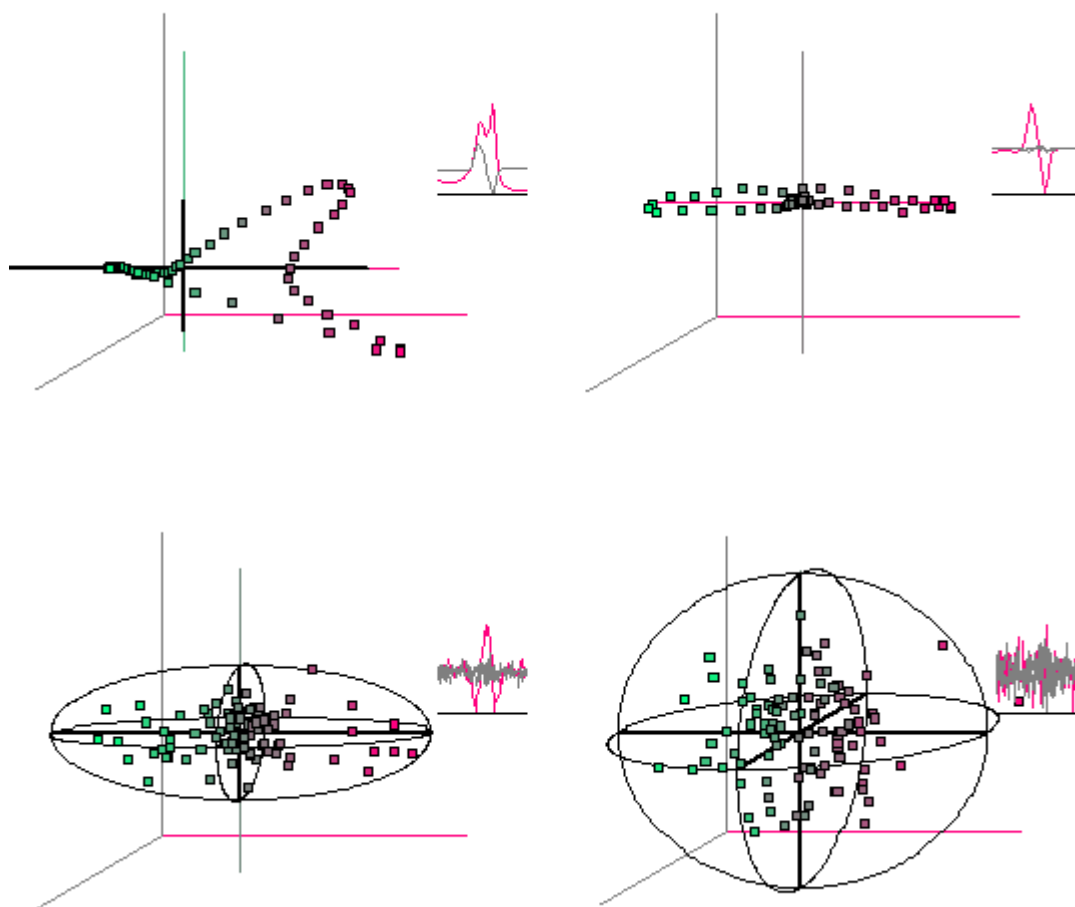


Figure 13: All six vectors are transformed by computing all six abstract factors. These data were selected to have three significant abstract factors. As a consequence the spherically symmetrical appearance of the bottom right tile is typical of abstract factors representative of noise. Mutual orthogonality is evidenced by the alignment of the axes in each scatter plot. Note how Figure 6 represents the state after computing the first abstract factor and how the scatter plots in Figure 12 do not align with the coordinate axes indicating the remaining vectors do not obey the conditions required for abstract factors.

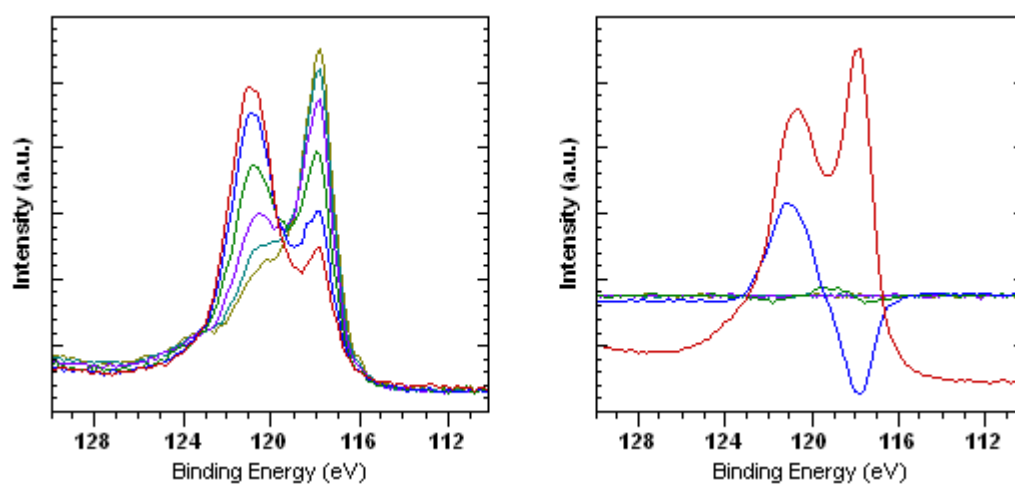


Figure 14: The six spectra used in Figures 7 through 13 shown as raw spectra and after all are transformed to abstract factor vectors corresponding to the scatter plots in Figure 13.