

## Path modelling

**Introduction.** The H-principle provides with a natural framework to analyse a network of data blocks. We shall here describe briefly the basic ideas and methods behind the path modelling procedures. For numerical details and a Matlab program, see [??].

The simplest case of path modelling is a standard regression analysis. Here we have two data blocks,  $\mathbf{X}$  and  $\mathbf{Y}$ . We write  $\mathbf{X} \Rightarrow \mathbf{Y}$  to indicate closer the path. It means that a new sample in  $\mathbf{X}$  generates an estimate of a new sample in  $\mathbf{Y}$ . When the  $\mathbf{Y}$ -sample is available, it can be compared with the estimated sample. We say that there is one input data block and one output. If we have three data blocks,  $\mathbf{X}$ ,  $\mathbf{Y}$  and  $\mathbf{Z}$ , we may have the path  $\mathbf{X} \Rightarrow \mathbf{Y} \Rightarrow \mathbf{Z}$ , which means that a new sample (a row) in  $\mathbf{X}$  generates an estimate of a new sample in  $\mathbf{Y}$ , which again generates an estimate of a new sample in  $\mathbf{Z}$ .  $\mathbf{X}$  is the input data block and  $\mathbf{Z}$  the output data block.

By symmetry the methods also contain projection along the columns. Thus,  $\mathbf{X} \Downarrow \mathbf{Y}$  is a projection of the rows of  $\mathbf{X}$  onto  $\mathbf{Y}$ . It is the same algorithm as the one above with  $\mathbf{X} \Rightarrow \mathbf{Y}$  but using the transpose of  $\mathbf{X}$  and  $\mathbf{Y}$ . The projection will show us how a new column of  $\mathbf{X}$  would generate a new column of  $\mathbf{Y}$ . When a new column of  $\mathbf{Y}$  becomes available it can be compared to the estimated column from  $\mathbf{X}$ .

The data blocks can make up a path of nodes, where each node contains two adjoining data blocks. A very general path or network of data blocks can be analysed. The only condition is that two neighbouring data blocks have either the same number of variables or the same number of samples. There can be arbitrary many input data blocks and output data blocks. The modelling task is to model input data blocks and generate regression models through the path to each of the output data blocks. Thus there is a regression model between each pair of data blocks that shows how the input samples propagate through that data block.

An important aspect of this approach is that most methods of linear regression analysis can be carried out within this framework, like e.g. sensitivity analysis, outlier detection, and dimension analysis. Also graphic tools, like score and loading plots, extend in a natural way. The procedures are based on projections of one latent structure onto the following one. These methods can therefore be used to detect (differential) changes in the latent structure (e.g. in loadings or scores) from one block to another.

The algorithms generate score and loading vectors for each data block. These are successive projections of previous loading and score

vectors. These score and loading vectors are connected together as specified by the NIPALS algorithm, developed by H. Wold, see Ref. [??]. We shall explain the NIPALS algorithm closer in case of standard linear regression.

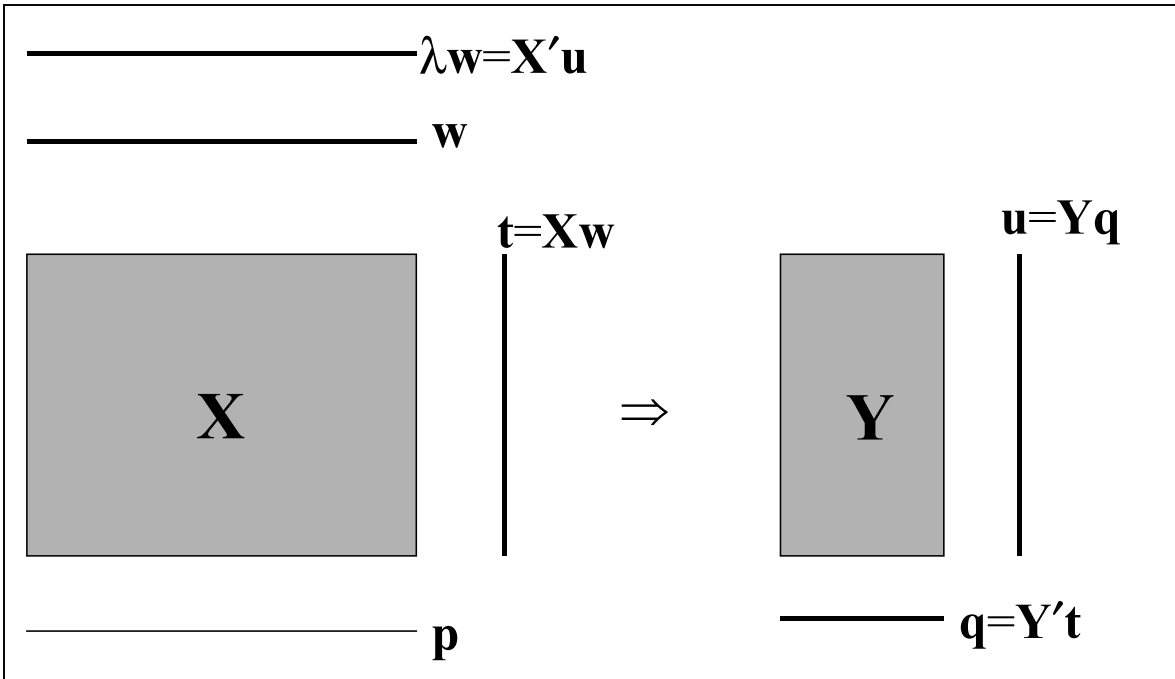
There is considerable interest in path modelling. In some part of applied sciences, e.g., social sciences, these methods are the dominant ones. In industry there is also great interest in these new methods. One data block can be the status of the production process and output data blocks e.g., the quality measures at the different stages of the production.

An important aspect of these methods is that they are computationally very fast. There can be numerous data blocks in the path, and we can define as starting data blocks any blocks anywhere in the path. Thus we can work with numerous regression models from different parts (data blocks) of the network to the output data blocks.

**Linear regression and the NIPALS algorithm.** Herman Wold, see Ref. [??], used the name NIPALS (Nonlinear Iterative PArtial Least Squares) for a common name of a sequence of projections that make up a closed loop. In the NIPALS algorithm we can start with a weight vector  $\mathbf{w}$ , scale it to unit length, then compute  $\mathbf{t}$  as  $\mathbf{t} = \mathbf{X}\mathbf{w}/(\mathbf{w}^T\mathbf{w}) = \mathbf{X}\mathbf{w}$ , a projection of  $\mathbf{X}$  onto  $\mathbf{w}$ . The score vector  $\mathbf{t}$  is then projected onto  $\mathbf{Y}$  to generate a  $\mathbf{Y}$ -loading vector  $\mathbf{q}$ . The loading vector then generates a  $\mathbf{Y}$ -score vector  $\mathbf{u}$  that is projected onto  $\mathbf{X}$ . The resulting vector is then used as a new weight vector  $\mathbf{w}$ . It is a property of the NIPALS algorithm that the closed loop projections, where scaling is e.g., done where we start, always will converge. It can be shown that the NIPLAS algorithm always leads to an eigen value system.

Let us now consider the recommendation of the H-principle. It suggests that we should find  $\mathbf{w}$  as the leading eigen vector of the eigen system  $\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X} \mathbf{w} = \lambda \mathbf{w}$ . It is easy to show that  $\mathbf{w}$  together with  $\mathbf{t}$ ,  $\mathbf{q}$  and  $\mathbf{u}$ ,  $\mathbf{t} = \mathbf{X}\mathbf{w}$ ,  $\mathbf{q} = \mathbf{Y}^T\mathbf{t}$ ,  $\mathbf{u} = \mathbf{Y}\mathbf{q}$ , form a NIPALS set of vectors with  $\lambda\mathbf{w} = \mathbf{X}^T\mathbf{u}$ . Each of the vectors is the eigen vector of an appropriate eigen system ( $\mathbf{X}\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T$ ,  $\mathbf{Y}^T\mathbf{X}\mathbf{X}^T\mathbf{Y}$  and  $\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{X}^T$ ). In the numerical computations one vector that is fastest to compute is found first and the others are computed from the NIPALS sequence. A schematic illustration of the NIPALS vectors is shown in Fig. 1.

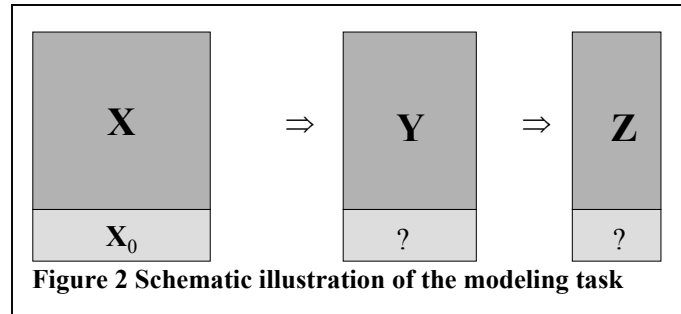
When the vectors have been found, the loading vector  $\mathbf{p}$  is found as  $\mathbf{p} = \mathbf{X}^T\mathbf{t}/(\mathbf{t}^T\mathbf{t})$ , and  $\mathbf{X}$  is adjusted (deflated) for the score and loading vector found,  $\mathbf{X} \leftarrow \mathbf{X} - \mathbf{t}\mathbf{p}^T$ . The data matrix  $\mathbf{Y}$  can be adjusted also,  $\mathbf{Y} \leftarrow \mathbf{Y} - \mathbf{t}\mathbf{q}^T/(\mathbf{t}^T\mathbf{t})$ . But this is not necessary, but can be useful, when carrying out significance testing. When  $\mathbf{X}$  has been adjusted a new iteration starts by computing a new set of NIPALS vectors.



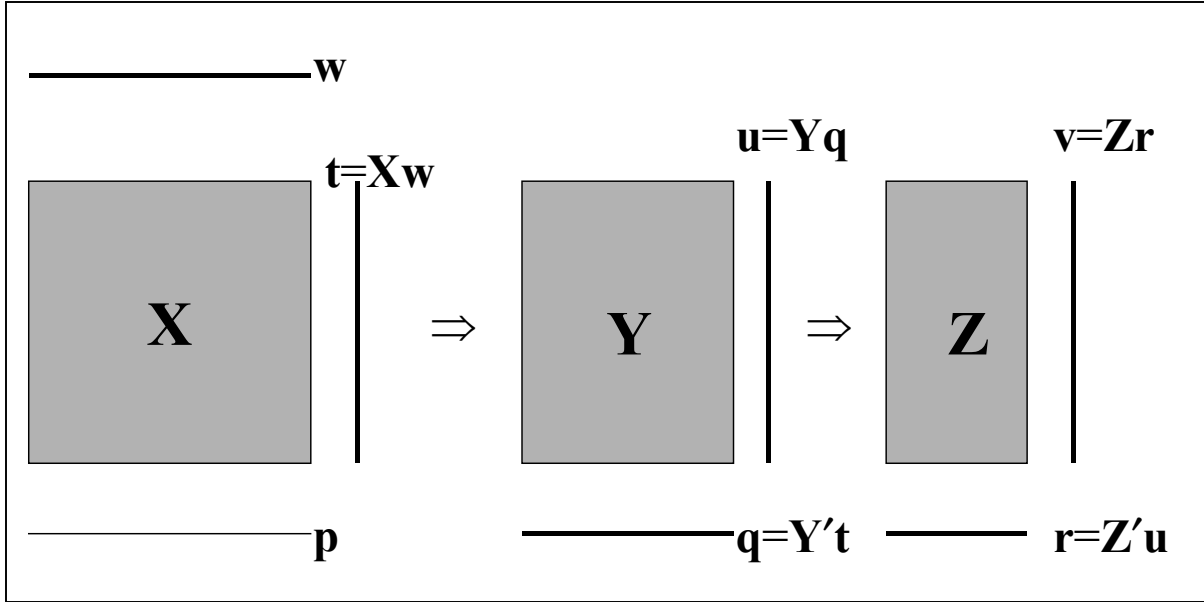
**Figure 1.** Schematic illustration of NIPALS vectors

In the graphic analysis the vectors  $\mathbf{t}$  and  $\mathbf{p}$  represent  $\mathbf{X}$ . The vectors  $\mathbf{u}$  and  $\mathbf{q}$  represent the projection onto  $\mathbf{Y}$ . Thus in studying  $\mathbf{X}$ , we look at the plots containing  $\mathbf{t}$  and  $\mathbf{p}$ , and for  $\mathbf{Y}$ ,  $\mathbf{u}$  and  $\mathbf{q}$ . The strength of the relationship between  $\mathbf{X}$  and  $\mathbf{Y}$  is studied by plotting  $\mathbf{u}$  against  $\mathbf{t}$ .

**Three data blocks.** We shall consider closer how the above ideas extend to three data blocks. We suppose that there is given three data blocks,  $\mathbf{X}$ ,  $\mathbf{Y}$  and  $\mathbf{Z}$ . It is assumed that they all have the same number of rows (samples). When new samples arrive at  $\mathbf{X}$ ,  $\mathbf{X}_0$ , we want to predict the values of the  $\mathbf{Y}$ - and  $\mathbf{Z}$ - samples. This is schematically illustrated in Fig. 2. Note, that we connect  $\mathbf{X}$  and  $\mathbf{Z}$  through  $\mathbf{Y}$ . When new samples of  $\mathbf{Y}$  become available, we can compare it to the predicted one. We can also use it in modeling  $\mathbf{Y} \Rightarrow \mathbf{Z}$ , which is developed separately. It is quite possible that  $\mathbf{Y}$  is spoiling the modeling task. It may be better to model  $\mathbf{X} \Rightarrow \mathbf{Z}$  directly. It is important in the path modeling to investigate the individual nodes (data blocks) to see if they function properly.



**Figure 2** Schematic illustration of the modeling task



**Figure 3.** Schematic illustration of NIPALS vectors in case of three data blocks.

In Fig. 3 we show a schematic illustration of the NIPALS vectors for the three data blocks. In case of two data blocks the optimization task was to maximize the size of the  $Y$ -loading vector  $q$ . A natural extension is to maximize the  $Z$ -loading vector  $r$ ,

$$\text{maximize } |r|^2 = \text{maximize } |Z^T Y Y^T X w|^2, \text{ subject to } |w|=1.$$

The solution is the leading eigen vector associated with the eigen system,

$$X^T Y Y^T Z Z^T Y Y^T X w = \lambda w.$$

When  $w$  has been found the other NIPALS vectors are computed as shown in Fig. 3. The  $X$ -loading vector  $p$  is computed and  $X$  adjusted in the same way as specified above. Then, a new iteration starts. The NIPALS vectors and  $p$  are used in the graphic studies of modelling task in a similar way as the case of two data blocks.

**Two input data blocks.** When we have two input (or starting) data blocks, the optimization procedure changes slightly. The H-principle suggests that we should be finding weight vectors  $w_1$  and  $w_2$ , such that the score vectors  $t_1=X_1 w_1$  and  $t_2=X_2 w_2$  have maximal covariance with  $Y$ . The task here is to find a weight vector  $p_y$  that together with the vectors  $w_1$  and  $w_2$  maximize the covariance,  $(t_3^T t_1)+(t_3^T t_2)$ , where  $t_3=Y p_y$ . The situation is schematically

illustrated in Fig. 4. In the maximization procedure we suppose that each of the weight vector has length one. The Lagrangian function of this maximization task is

$$f(\mathbf{w}_1, \mathbf{w}_2, \mathbf{p}_y) = \mathbf{w}_1^T \mathbf{X}_1^T \mathbf{Y} \mathbf{p}_y + \mathbf{w}_2^T \mathbf{X}_2^T \mathbf{Y} \mathbf{p}_y - \lambda_1 (\mathbf{w}_1^T \mathbf{w}_1 - 1) - \lambda_2 (\mathbf{w}_2^T \mathbf{w}_2 - 1) - \lambda_3 (\mathbf{p}_y^T \mathbf{p}_y - 1)$$

If we differentiate the function  $f$  with respect to  $\mathbf{w}_1$ ,  $\mathbf{w}_2$  and  $\mathbf{p}_y$ , we get

$$\begin{aligned} \mathbf{X}_1^T \mathbf{Y} \mathbf{p}_y - 2\lambda_1 \mathbf{w}_1 &= \mathbf{0} \\ \mathbf{X}_2^T \mathbf{Y} \mathbf{p}_y - 2\lambda_2 \mathbf{w}_2 &= \mathbf{0} \\ \mathbf{Y}^T \mathbf{X}_1 \mathbf{w}_1 + \mathbf{Y}^T \mathbf{X}_2 \mathbf{w}_2 - 2\lambda_3 \mathbf{p}_y &= \mathbf{0} \end{aligned}$$

If we solve with respect to  $\mathbf{p}_y$ , we get the following set of equations for  $\mathbf{w}_1$  and  $\mathbf{w}_2$ ,

$$\begin{aligned} \mathbf{X}_1^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}_1 \mathbf{w}_1 + \mathbf{X}_1^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}_2 \mathbf{w}_2 &= \mu_1 \mathbf{w}_1 \\ \mathbf{X}_2^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}_1 \mathbf{w}_1 + \mathbf{X}_2^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}_2 \mathbf{w}_2 &= \mu_2 \mathbf{w}_2 \end{aligned}$$

Here  $\mu_1$  and  $\mu_2$  are derived from the  $\lambda$ 's. These equations show that the  $\mathbf{w}$ 's are found as eigen vectors of the patterned covariance matrix,  $(\mathbf{X}_i^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}_j)$ . Note that if e.g.,  $\mathbf{t}_2$  does not contribute to the analysis it is dropped.

Some small changes in the optimization procedure can be recommended, when the input data blocks are of different types. Suppose that  $\mathbf{X} \Rightarrow \mathbf{Y}$  and  $\mathbf{Z} \hat{=} \mathbf{Y}$ . The situation is schematically illustrated in Fig. 5. If we look at  $\mathbf{X} \Rightarrow \mathbf{Y}$  only, we should find  $\mathbf{w}$  such that the size of  $\mathbf{q}$  is as large as possible. On the other hand, if we only look at  $\mathbf{Z} \hat{=} \mathbf{Y}$ , we should find  $\mathbf{v}$  such that  $\mathbf{u}$  is as large as possible. If we carry out these two optimization tasks independently of each other, the data block  $\mathbf{Y}$  is represented by the vectors  $\mathbf{q}$  and  $\mathbf{u}$ . The rank one matrix representing the projection onto  $\mathbf{Y}$  is given by  $\mathbf{u} \mathbf{q}^T / (\mathbf{t}^T \mathbf{Y} \mathbf{r})$ . The numerical problem that easily can arise is that  $\mathbf{t}^T \mathbf{Y} \mathbf{r} \approx 0$ . In that case there is numerical instability in the computations. The way to

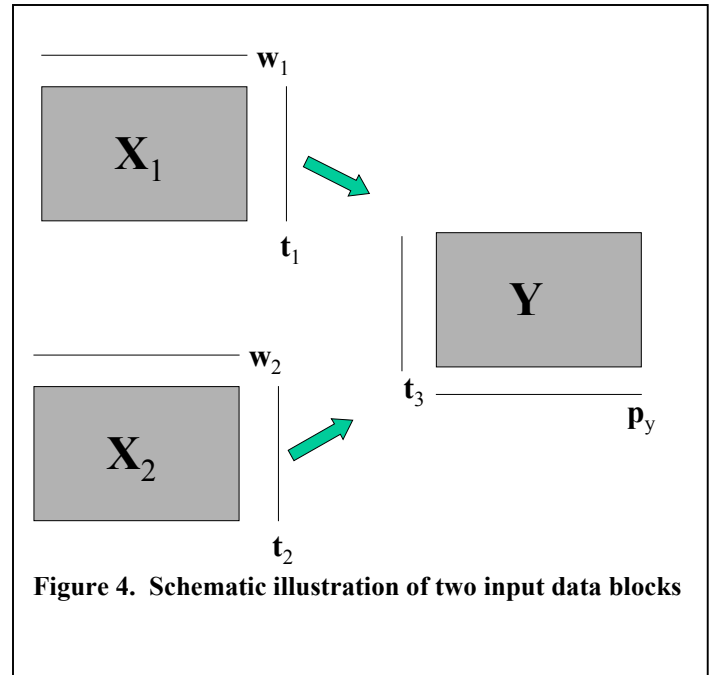


Figure 4. Schematic illustration of two input data blocks

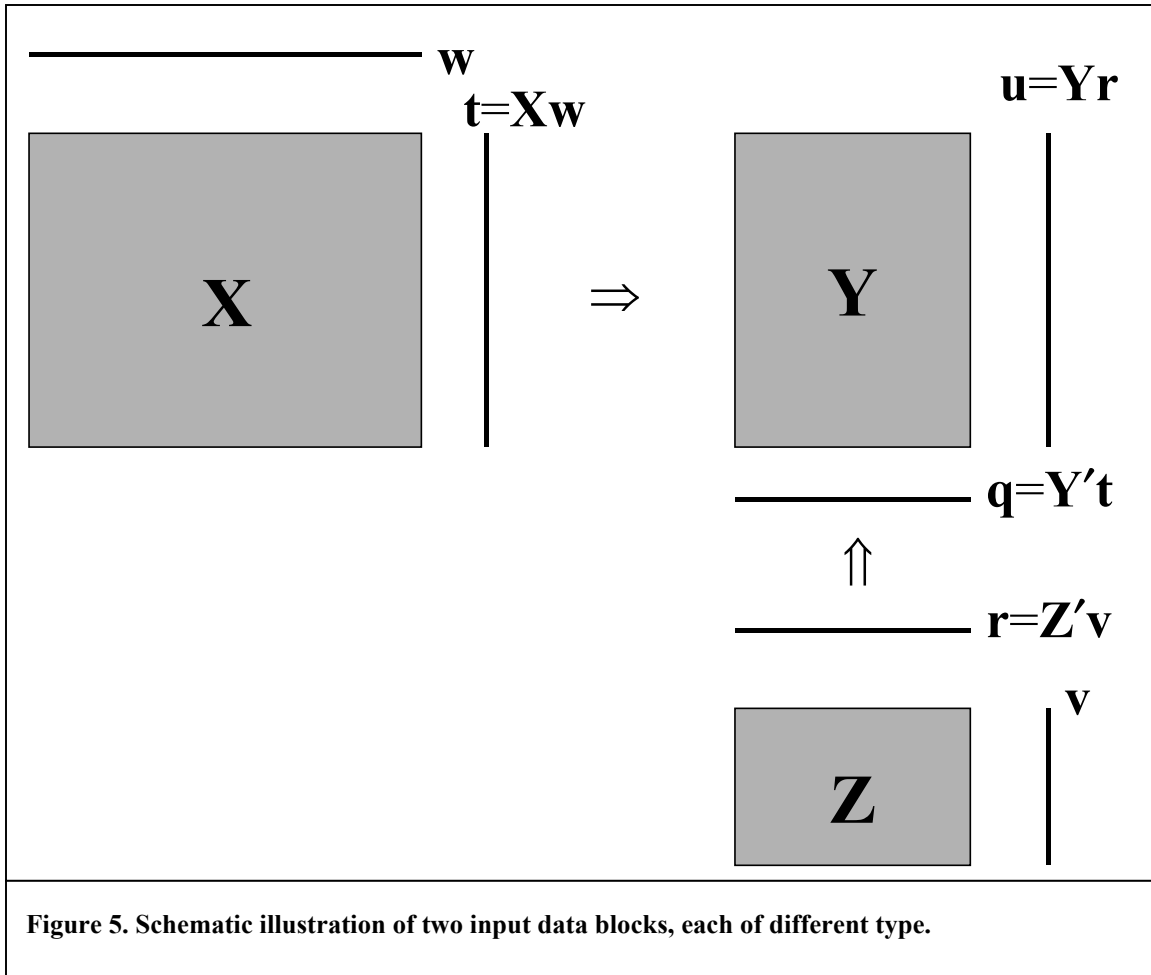


Figure 5. Schematic illustration of two input data blocks, each of different type.

secure numerical stability is to have only one optimization task. Analogous to the procedure above we define a  $\mathbf{Y}$ -score vector as  $\mathbf{t}_3=\mathbf{Y}\mathbf{p}_y$ . We require that the score vector generated by  $\mathbf{X}$ ,  $\mathbf{t}=\mathbf{X}\mathbf{w}$ , and the score vector generated by  $\mathbf{Z}$ ,  $\mathbf{u}$ , to have maximal covariance with  $\mathbf{t}_3$ . Thus we are to find  $\mathbf{w}$ ,  $\mathbf{v}$  and  $\mathbf{p}_y$  such that  $(\mathbf{t}_3^T\mathbf{t})+(\mathbf{t}_3^T\mathbf{u})$ , is maximized. The Lagrangian function of this maximisation task is

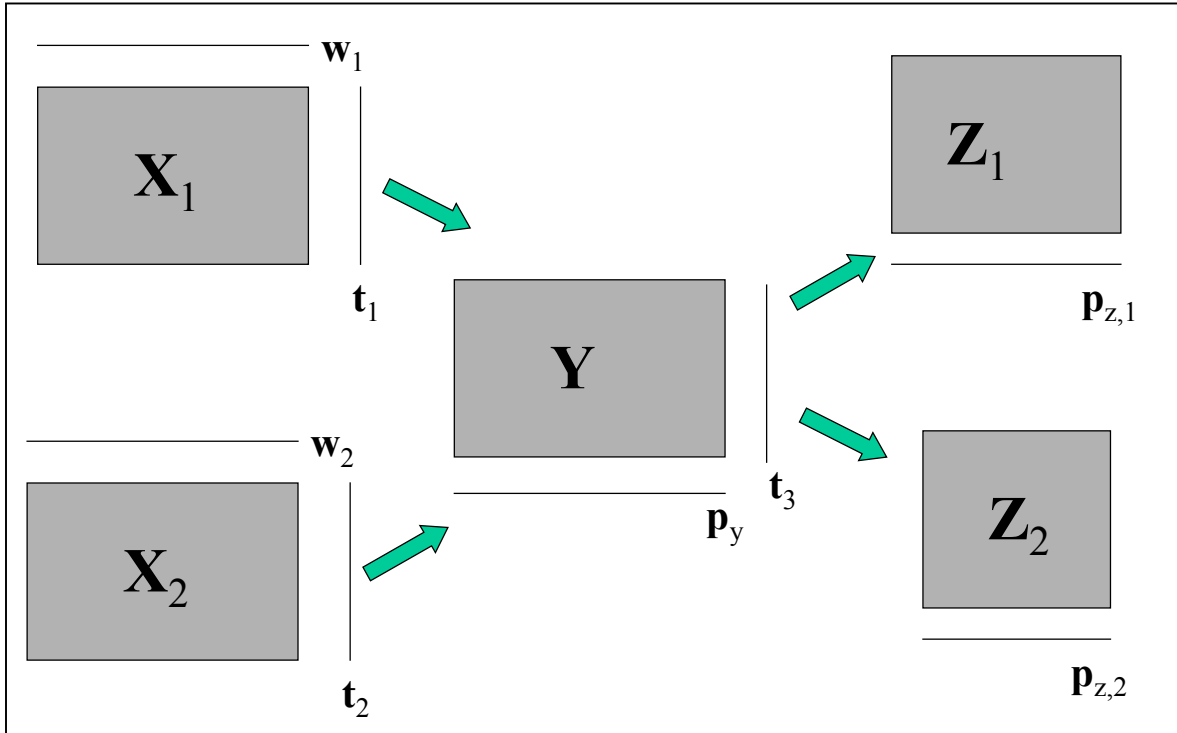
$$f(\mathbf{w}, \mathbf{v}, \mathbf{p}_y) = \mathbf{w}^T \mathbf{X}^T \mathbf{Y} \mathbf{p}_y + \mathbf{v}^T \mathbf{Z} \mathbf{Y}^T \mathbf{Y} \mathbf{p}_y - \lambda_1(\mathbf{w}^T \mathbf{w} - 1) - \lambda_2(\mathbf{v}^T \mathbf{v} - 1) - \lambda_3(\mathbf{p}_y^T \mathbf{p}_y - 1)$$

A similar procedure as above leads to the set of eigen systems,

$$\begin{aligned} \mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X} \mathbf{w} + \mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{Y} \mathbf{Z}^T \mathbf{w} &= \mu_1 \mathbf{w} \\ \mathbf{Z} \mathbf{Y}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X} \mathbf{w} + \mathbf{Z} \mathbf{Y}^T \mathbf{Y} \mathbf{Y}^T \mathbf{Y} \mathbf{Z}^T \mathbf{v} &= \mu_2 \mathbf{v}, \end{aligned}$$

where  $\mathbf{X}_2$  above is replaced by  $\mathbf{Y} \mathbf{Z}^T$ . This is solved similarly. Note that instead of working with the score vectors, we could work with the loading vectors.

**Two input and two output data blocks.** We shall look closer at the situation, where we have two input and two output data blocks. The situation is schematically illustrated in Fig. 6. We have a weight vector for each input data block. The task is to find optimal weight vectors. Like above we want the resulting loading vectors at the output to be as large as possible. Let us look closer how we find the weight vectors.



**Figure 6.** Schematic illustration of the modelling task including two input and two output data blocks.

In analogy with the criteria above we want the sizes of  $\mathbf{p}_{z,1}$  and  $\mathbf{p}_{z,2}$  to be as large as possible. Let  $\mathbf{p}=(\mathbf{p}_{z,1}, \mathbf{p}_{z,2})$ . Then we can write

$$|\mathbf{p}|^2 = |\mathbf{p}_{z,1}|^2 + |\mathbf{p}_{z,2}|^2 = |\mathbf{Z}_1^T \mathbf{t}_3|^2 + |\mathbf{Z}_2^T \mathbf{t}_3|^2 = |(\mathbf{Z}_1, \mathbf{Z}_2)^T \mathbf{t}_3|^2 = |\mathbf{Z}^T \mathbf{t}_3|^2$$

with  $\mathbf{Z}=(\mathbf{Z}_1, \mathbf{Z}_2)$ . The loading vectors at the end of the path are computed as

$$\mathbf{p}_{z,1} = \mathbf{Z}_1^T \mathbf{Y} \mathbf{Y}^T (\mathbf{X}_1 \mathbf{w}_1 + \mathbf{X}_2 \mathbf{w}_2) \quad \text{and} \quad \mathbf{p}_{z,2} = \mathbf{Z}_2^T \mathbf{Y} \mathbf{Y}^T (\mathbf{X}_1 \mathbf{w}_1 + \mathbf{X}_2 \mathbf{w}_2).$$

If we want to find  $\mathbf{w}_1$  and  $\mathbf{w}_2$  that maximise the total (squared) size of the loadings,  $|\mathbf{p}_{z,1}|^2 + |\mathbf{p}_{z,2}|^2$ , it is easy to show using the Lagrangian multiplier technique that the weight vectors are found as solutions to the eigen system,

$$\mathbf{X}_1^T \mathbf{Y} \mathbf{Y}^T \mathbf{Z} \mathbf{Z}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}_1 \mathbf{w}_1 + \mathbf{X}_1^T \mathbf{Y} \mathbf{Y}^T \mathbf{Z} \mathbf{Z}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}_2 \mathbf{w}_2 = \lambda_1 \mathbf{w}_1$$

$$\mathbf{X}_1^T \mathbf{Y} \mathbf{Y}^T \mathbf{Z} \mathbf{Z}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}_2 \mathbf{w}_1 + \mathbf{X}_2^T \mathbf{Y} \mathbf{Y}^T \mathbf{Z} \mathbf{Z}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}_2 \mathbf{w}_2 = \lambda_2 \mathbf{w}_2$$

When we extend these considerations to numerous input data matrices, we notice that the  $\mathbf{w}$ 's are found the eigen vectors of the patterned covariance matrix,  $(\mathbf{X}_i^T \mathbf{Y} \mathbf{Y}^T \mathbf{Z} \mathbf{Z}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}_j)$ . Similarly, when we have output data blocks scattered in the network, we get analogous patterned covariance matrices.

**Units of paths.** The paths that are allowed can be very general. The following table shows possible 'units' in paths.

	<b>Paths</b>	<b>Interpretation</b>
1	$\mathbf{X}$	One block. We get PCA or PCA-types of solutions.
2	$\mathbf{X}_1 \Rightarrow \mathbf{X}_2$	Two blocks. We get linear regression.
3	$\mathbf{X}_1 \Rightarrow \mathbf{X}_2 \Rightarrow \mathbf{X}_3$	Multi-block extensions of linear regression.
4	$\mathbf{X}_1$ $\Downarrow$ $\mathbf{X}_2$	Multi-block extensions of PCA-type of solutions with the role of variables and samples exchanged.
5	$\mathbf{X}_2 \Rightarrow \mathbf{X}_3$ $\Uparrow$ $\mathbf{X}_1$	Here we want the regression to be done with components generated from $\mathbf{X}_1$ .
6	$\mathbf{X}_1 \Rightarrow \mathbf{X}_2$ $\Downarrow$ $\mathbf{X}_3$	Here we want to study possible changes in the response values even if we have not been able to observe or measure the corresponding X-values.
7	$\mathbf{X}_1 \Rightarrow \mathbf{X}_2$ $\Uparrow$ $\mathbf{X}_3$	Here are two 'sources' $\mathbf{X}_1$ and $\mathbf{X}_3$ that influence on the data block $\mathbf{X}_2$ .

These are a few possible parts of paths that the algorithms can carry out. These models can be combined arbitrarily into a large and complex network. The only requirement is that the number of samples/variables in two succeeding blocks,  $\mathbf{X}_k$  and  $\mathbf{X}_{k+1}$ , are correct. If  $\mathbf{X}_k \Rightarrow \mathbf{X}_{k+1}$ , the number of samples (rows) in  $\mathbf{X}_k$  and  $\mathbf{X}_{k+1}$ , is the same, while if  $\mathbf{X}_k \Downarrow \mathbf{X}_{k+1}$ , the number of



variables (columns) must be equal. Note that type 5 in the table can be interpreted as that we want to be able to predict values in  $\mathbf{X}_3$  from observing samples of type  $\mathbf{X}_1$ . It is an example of making predictions of response variables although we are not able to measure the response variables.

It is an important aspect of the methods that if we have one input and one output data matrix, and all matrices are equal,  $\mathbf{X}_1=\mathbf{X}_2=\mathbf{X}_3=\dots$ , we get the PCA solution for each data matrix. Thus, when there are small changes between the data blocks, the score vectors found will describe the changes in the samples and the loading vectors the changes in the variables.