Institute of Phonetic Sciences,
University of Amsterdam,
Proceedings 25 (2003), 81–99.

CANONICAL CORRELATION ANALYSIS

David Weenink

Abstract

We discuss algorithms for performing canonical correlation analysis. In canonical correlation analysis we try to find correlations between two data sets. The canonical correlation coefficients can be calculated directly from the two data sets or from (reduced) representations such as the covariance matrices. The algorithms for both representations are based on singular value decomposition. The methods described here have been implemented in the speech analysis program PRAAT (Boersma & Weenink, 1996), and some examples will be demonstated for formant frequency and formant level data from 50 male Dutch speakers as were reported by Pols et al. (1973).

1 Introduction

Let $\mathbf{X}$ be a data matrix of dimension $m \times n$ which contains m representations of an n-dimensional vector of random variables $\mathbf{x}$. The correlation coefficient ρ_{ij} that shows the correlation between the variables x_i and x_j is defined as

$$\rho_{ij} = \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii}\Sigma_{jj}}}, \tag{1}$$

where the number Σ_{ij} denotes the covariance between x_i and x_j which is defined as

$$\Sigma_{ij} = \frac{1}{m-1}\sum_{k=1}^{m}(X_{ki} - \mu_i)(X_{kj} - \mu_j), \tag{2}$$

where μ_i is x_i's average value. The matrix Σ is called the covariance matrix. From $\mathbf{X}$ we construct the data matrix $\mathbf{A}_x$ by centering the columns of $\mathbf{X}$, i.e., the elements of $\mathbf{A}_x$, the a_{ij}, are $a_{ij} = X_{ij} - \mu_j$. We can now rewrite the covariance matrix as

$$\Sigma = \frac{1}{m-1}\mathbf{A}_x'\mathbf{A}_x, \tag{3}$$

where $\mathbf{A}_x'$ denotes the transpose of $\mathbf{A}_x$.

Note that the correlation coefficient only provides a measure of the *linear* association between the two variables: when the two variables are uncorrelated, i.e., when their correlation coefficient is zero, this only means that no linear function describes their relationship. A quadratic relationship or some other non-linear relationship is certainly not ruled out.

Equation (1) shows us the recipe to determine the correlation matrix from the covariance matrix. However, the correlations in the correlation matrix depend very much

on the coordinate system that we happen to use. We could rotate the coordinate system in such a way that the projections in the new coordinate system are maximally uncorrelated and this is exactly what a principal component analysis does achieve: the correlation matrix obtained from the principal components would be the identity matrix, showing only zeros with ones on the diagonal. While each element in the correlation matrix captures the correlation between two variables, the object of canonical correlation analysis is to capture the correlations between two *sets* of variables. Canonical correlation analysis tries to find basis vectors for two sets of multidimensional variables such that the linear correlations between the projections onto these basis vectors are mutually maximized. In the limit when the dimension of each set is 1, the canonical correlation coefficient reduces to the correlation coefficient.

We will need this type of analysis when we want to find relations between different representations of the same objects. In here we will demonstrate its usefulness by showing, for example, the correlations between principal components and auto-associative neural nets for vowel data.

2 Mathematical background

Canonical correlation analysis originates in Hotelling (1936) and the two equations that govern the analysis are the following:

$$(\Sigma'_{xy}\Sigma_{xx}^{-1}\Sigma_{xy} - \rho^2\Sigma_{yy})\mathbf{y} = \mathbf{0} \quad (4)$$

$$(\Sigma_{xy}\Sigma_{yy}^{-1}\Sigma'_{xy} - \rho^2\Sigma_{xx})\mathbf{x} = \mathbf{0}, \quad (5)$$

where Σ'_{xy} denotes the transpose of Σ_{xy}. Both equations look similar and have, in fact, the same eigenvalues. And, given the eigenvectors for one of these equations, we can deduce the eigenvectors for the other as will be shown in the next section.

2.1 Derivation of the canonical correlation analysis equations

In canonical correlation analysis we want to maximize correlations between objects that are represented with two data sets. Let these data sets be $\mathbf{A}_x$ and $\mathbf{A}_y$, of dimensions $m \times n$ and $m \times p$, respectively. Sometimes the data in $\mathbf{A}_y$ and $\mathbf{A}_x$ are called the *dependent* and the *independent* data, respectively. The maximum number of correlations that we can find is then equal to the minimum of the column dimensions n and p. Let the directions of optimal correlations for the $\mathbf{A}_x$ and $\mathbf{A}_y$ data sets be given by the vectors $\mathbf{x}$ and $\mathbf{y}$, respectively. When we project our data on these direction vectors, we obtain two new vectors $\mathbf{z}_x$ and $\mathbf{z}_y$, defined as follows:

$$\mathbf{z}_x = \mathbf{A}_x\mathbf{x} \quad (6)$$

$$\mathbf{z}_y = \mathbf{A}_y\mathbf{y}. \quad (7)$$

The variables $\mathbf{z}_y$ and $\mathbf{z}_x$ are called the *scores* or the *canonical variates*. The correlation between the scores $\mathbf{z}_y$ and $\mathbf{z}_x$ is then given by:

$$\rho = \frac{\mathbf{z}'_y \cdot \mathbf{z}_x}{\sqrt{\mathbf{z}'_y \cdot \mathbf{z}_y}\sqrt{\mathbf{z}'_x \cdot \mathbf{z}_x}}. \quad (8)$$

Our problem is now finding the directions $\mathbf{y}$ and $\mathbf{x}$ that maximize equation (8) above. We first note that ρ is not affected by a rescaling of $\mathbf{z}_y$ or $\mathbf{z}_x$, i.e., a multiplication of

$\mathbf{z}_y$ by the scalar α does not change the value of ρ in equation (8). Since the choice of rescaling is arbitrary, we therefor maximize equation (8) subject to the constraints

$$\mathbf{z}'_x \cdot \mathbf{z}_x = \mathbf{x}'\mathbf{A}'_x\mathbf{A}_x\mathbf{x} = \mathbf{x}'\Sigma_{xx}\mathbf{x} = 1 \tag{9}$$

$$\mathbf{z}'_y \cdot \mathbf{z}_y = \mathbf{y}'\mathbf{A}'_y\mathbf{A}_y\mathbf{y} = \mathbf{y}'\Sigma_{yy}\mathbf{y} = 1. \tag{10}$$

We have made the substitutions $\Sigma_{yy} = \mathbf{A}'_y\mathbf{A}_y$ and $\Sigma_{xx} = \mathbf{A}'_x\mathbf{A}_x$, where the Σ's are covariance matrices (the scaling factor to get the covariance matrix, $1/(m-1)$, can be left out without having any influence on the result). When we also substitute $\Sigma_{yx} = \mathbf{A}'_y\mathbf{A}_x$ we use the two constraints above and write the maximization problem in Lagrangian form:

$$L(\rho_x, \rho_y, \mathbf{x}, \mathbf{y}) = \mathbf{y}'\Sigma_{yx}\mathbf{x} - \frac{\rho_x}{2}\left(\mathbf{x}'\Sigma_{xx}\mathbf{x} - 1\right) - \frac{\rho_y}{2}\left(\mathbf{y}'\Sigma_{yy}\mathbf{y} - 1\right), \tag{11}$$

We can solve equation (11) by first taking derivatives with respect to $\mathbf{y}$ and $\mathbf{x}$:

$$\frac{\partial L}{\partial \mathbf{x}} = \Sigma_{xy}\mathbf{y} - \rho_x\Sigma_{xx}\mathbf{x} = \mathbf{0} \tag{12}$$

$$\frac{\partial L}{\partial \mathbf{y}} = \Sigma_{yx}\mathbf{x} - \rho_y\Sigma_{yy}\mathbf{y} = \mathbf{0}. \tag{13}$$

Now subtract $\mathbf{x}'$ times the first equation from $\mathbf{y}'$ times the second and we have

$$\begin{aligned}\mathbf{0} &= \mathbf{y}'\Sigma_{yx}\mathbf{x} - \rho_y\mathbf{y}'\Sigma_{yy}\mathbf{y} - \mathbf{x}'\Sigma_{xy}\mathbf{y} + \rho_x\mathbf{x}'\Sigma_{xx}\mathbf{x} \\ &= \rho_x\mathbf{x}'\Sigma_{xx}\mathbf{x} - \rho_y\mathbf{y}'\Sigma_{yy}\mathbf{y}.\end{aligned}$$

Together with the constraints of equations (9) and (10) we must conclude that $\rho_x = \rho_y = \rho$. When Σ_{xx} is invertible we get from (12)

$$\mathbf{x} = \frac{\Sigma_{xx}^{-1}\Sigma_{xy}\mathbf{y}}{\rho}. \tag{14}$$

Substitution in (13) gives after rearranging essentially equation (4):

$$(\Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy} - \rho^2\Sigma_{yy})\mathbf{y} = \mathbf{0}. \tag{15}$$

In an analogous way we can get the equation for the vectors $\mathbf{x}$ as:

$$(\Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx} - \rho^2\Sigma_{xx})\mathbf{x} = \mathbf{0}. \tag{16}$$

Because the matrices Σ_{xy} and Σ_{yx} are each other's transpose we write the canonical correlation analysis equations as follows

$$(\Sigma'_{xy}\Sigma_{xx}^{-1}\Sigma_{xy} - \rho^2\Sigma_{yy})\mathbf{y} = \mathbf{0} \tag{17}$$

$$(\Sigma_{xy}\Sigma_{yy}^{-1}\Sigma'_{xy} - \rho^2\Sigma_{xx})\mathbf{x} = \mathbf{0}. \tag{18}$$

We can now easily see that in the one-dimensional case both equations reduce to a squared form of equation (1). The equations (17) and (18) are so called generalized eigenvalue problems. Special software is needed to solve these equations in a numerically stable and robust manner. In the next section we will discuss two methods to solve these equations. Both methods have been implementend in the PRAAT program.

2.2 Solution of the canonical correlation analysis equations

We can consider two cases here: the simple case when we only have the covariance matrices, or, the somewhat more involved case, when we have the original data matrices at our disposal.

2.2.1 Solution from covariance matrices

We will start with the simple case and solve equations (17) and (18) when we have the covariance matrices Σ_{xx}, Σ_{xy} and Σ_{yy} at our disposal. We will solve one equation and show that the solution for the second equation can be calculated from it. Provided Σ_{yy} is not singular, a simpler looking equation can be obtained by multiplying equation (17) from the left by Σ_{yy}^{-1}:

$$(\Sigma_{yy}^{-1}\Sigma_{xy}'\Sigma_{xx}^{-1}\Sigma_{xy} - \rho^2)\mathbf{y} = \mathbf{0}. \tag{19}$$

This equation can be solved in two steps. First we perform the two matrix inversions and the three matrix multiplications. In the second step we solve for the eigenvalues and eigenvectors of the resulting general square matrix. From the standpoint of numerical precision, actually performing the matrix inversions and multiplications, would be a very unwise thing to do because with every matrix multiplication we loose numerical precision. Instead of solving equation (17) with the method described above, we will rewrite this generalized *eigenvalue* problem as a generalized *singular value* problem. To accomplish this we will need the Cholesky factorization of the two symmetric matrices Σ_{xx} and Σ_{yy}.

The Cholesky factorization can be performed on symmetric positive definite matrices, like covariance matrices, and is numerically very stable (Golub & van Loan, 1996). Here we factor the covariance matrices as follows

$$\begin{aligned}\Sigma_{yy} &= \mathbf{U}_y'\mathbf{U}_y \\ \Sigma_{xx} &= \mathbf{U}_x'\mathbf{U}_x,\end{aligned}$$

where $\mathbf{U}_y$ and $\mathbf{U}_x$ are upper triangular matrices with positive diagonal entries. Let $\mathbf{K}$ be the inverse of $\mathbf{U}_x$, then we can write

$$\Sigma_{xx}^{-1} = \mathbf{K}\mathbf{K}'. \tag{20}$$

We substitute this in equation (17) and rewrite as

$$((\mathbf{K}'\Sigma_{xy})'(\mathbf{K}'\Sigma_{xy}) - \rho^2\mathbf{U}_y'\mathbf{U}_y)\mathbf{y} = \mathbf{0}. \tag{21}$$

This equation is of the form $(\mathbf{A}'\mathbf{A} - \rho\mathbf{B}'\mathbf{B})\mathbf{x} = \mathbf{0}$ which can be solved by a numerically very stable generalized singular value decomposition of $\mathbf{A}$ and $\mathbf{B}$, without actually performing the matrix multiplications $\mathbf{A}'\mathbf{A}$ and $\mathbf{B}'\mathbf{B}$ (Golub & van Loan, 1996; Weenink, 1999). We have obtained this equation by only one matrix multiplication, two Cholesky decompositions and one matrix inversion. This allows for a better estimation of the eigenvalues than estimating them from equation (19). The square roots of the eigenvalues of equation (21) are the canonical correlation coefficients ρ. The eigenvectors $\mathbf{y}$ tell us how to combine the columns of $\mathbf{A}_y$ to get this optimum canonical correlation.

We will now show that the eigenvalues of equations (17) and (18) are equal and that the eigenvectors for the latter can be obtained from the eigenvectors of the former. We first multiply (17) from the left by $\Sigma_{xy}\Sigma_{yy}^{-1}$ and obtain

$$(\Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}'\Sigma_{xx}\Sigma_{xy} - \rho^2\Sigma_{xy})\mathbf{y} = \mathbf{0},$$

which can be rewritten by inserting the identity matrix $\Sigma_{xx}\Sigma_{xx}^{-1}$ as

$$(\Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}'\Sigma_{xx}^{-1}\Sigma_{xy} - \rho^2\Sigma_{xx}\Sigma_{xx}^{-1}\Sigma_{xy})\mathbf{y} = \mathbf{0}.$$

Finally we split off the common $\Sigma_{xx}^{-1}\Sigma_{xy}$ part on the right and obtain

$$(\Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}' - \rho^2\Sigma_{xx})\Sigma_{xx}^{-1}\Sigma_{xy}\mathbf{y} = \mathbf{0}. \tag{22}$$

We have now obtained equation (18). This shows that the eigenvalues of equations (17) and (18) are equal and that the eigenvectors $\mathbf{x}$ for equation (18) can be obtained from the eigenvectors $\mathbf{y}$ of equation (17) as $\mathbf{x} = \Sigma_{xx}^{-1}\Sigma_{xy}\mathbf{y}$. This relation between the eigenvectors was already explicit in equation (14).

2.2.2 Solution from data matrices

When we have the data matrices $\mathbf{A}_x$ and $\mathbf{A}_y$ at our disposal we do not need to calculate the covariance matrices $\Sigma_{xx} = \mathbf{A}'\mathbf{A}_x$, $\Sigma_{yy} = \mathbf{A}_y'\mathbf{A}_y$ and $\Sigma_{xy} = \mathbf{A}_x'\mathbf{A}_y$ from them. Numerically spoken, there are better ways to solve equations (4) and (5). We will start with the singular value decompositions

$$\mathbf{A}_x = \mathbf{U}_x\mathbf{D}_x\mathbf{V}_x' \tag{23}$$
$$\mathbf{A}_y = \mathbf{U}_y\mathbf{D}_y\mathbf{V}_y' \tag{24}$$

and use them to obtain the following covariance matrices

$$\begin{aligned}\Sigma_{xx} &= \mathbf{A}_x'\mathbf{A}_x = \mathbf{V}_x\mathbf{D}_x^2\mathbf{V}_x'\\ \Sigma_{yy} &= \mathbf{A}_y'\mathbf{A}_y = \mathbf{V}_y\mathbf{D}_y^2\mathbf{V}_y'\\ \Sigma_{xy} &= \mathbf{A}_x'\mathbf{A}_y = \mathbf{V}_x\mathbf{D}_x\mathbf{U}_x'\mathbf{U}_y\mathbf{D}_y\mathbf{V}_y'.\end{aligned} \tag{25}$$

We use these decompositions together with $\Sigma_{xx}^{-1} = \mathbf{V}_x\mathbf{D}_x^{-2}\mathbf{V}_x'$ to rewrite equation (4) as

$$(\mathbf{V}_y\mathbf{D}_y\mathbf{U}_y'\mathbf{U}_x\mathbf{U}_x'\mathbf{U}_y\mathbf{D}_y\mathbf{V}_y' - \rho^2\mathbf{V}_y\mathbf{D}_y^2\mathbf{V}_y')\mathbf{y} = \mathbf{0}, \tag{26}$$

where we used the orthogonalities $\mathbf{V}_x'\mathbf{V}_x = \mathbf{I}$ and $\mathbf{V}_y'\mathbf{V}_y = \mathbf{I}$. Next we multiply from the left with $\mathbf{D}_y^{-1}\mathbf{V}_y'$ and obtain

$$(\mathbf{U}_y'\mathbf{U}_x\mathbf{U}_x'\mathbf{U}_y\mathbf{D}_y\mathbf{V}_y' - \rho^2\mathbf{D}_y\mathbf{V}_y')\mathbf{y} = \mathbf{0}, \tag{27}$$

which can be rewritten as

$$((\mathbf{U}_x'\mathbf{U}_y)'(\mathbf{U}_x'\mathbf{U}_y) - \rho^2\mathbf{I})\mathbf{D}_y\mathbf{V}_y'\mathbf{y} = \mathbf{0}. \tag{28}$$

This equation is of the form $(\mathbf{A}'\mathbf{A} - \rho\mathbf{I})\mathbf{x} = \mathbf{0}$ which can be easily solved by the substitution of the singular value decomposition (svd) of $\mathbf{A}$. The svd of $\mathbf{U}_x'\mathbf{U}_y = \mathbf{U}\mathbf{D}\mathbf{V}'$ substituted in equation (28) leaves us after some rearrangement with

$$(\mathbf{D}^2 - \rho^2\mathbf{I})\mathbf{V}'\mathbf{D}_y\mathbf{V}_y'\mathbf{y} = \mathbf{0}. \tag{29}$$

This equation has eigenvalues $\mathbf{D}^2$ and the eigenvectors can be obtained from the columns of $\mathbf{V}_y\mathbf{D}_y^{-1}\mathbf{V}$. In an analogous way we can reduce equation (5) to

$$(\mathbf{D}^2 - \rho^2\mathbf{I})\mathbf{U}'\mathbf{D}_x\mathbf{V}_x'\mathbf{x} = \mathbf{0}, \tag{30}$$

with the same eigenvalues $\mathbf{D}^2$. Analogously, the eigenvectors are obtained from the columns of $\mathbf{V}_x\mathbf{D}_x^{-1}\mathbf{U}$.

We now have shown that the algorithms above significantly reduce the number of matrix multiplications that are necessary to obtain the eigenvalues. First of all we do not actually need to perform the matrix multiplications to obtain the covariance matrices in equations (25). We only need two singular value decompositions and one matrix multiplication $\mathbf{U}_x'\mathbf{U}_y$. The latter multiplication is numerically very stable because both matrices are column orthogonal.

2.2.3 Solution summary

We have shown two numerically stable procedures to solve the canonical correlation equations (4) and (5). In both procedures the data matrices $\mathbf{A}_x$ and $\mathbf{A}_y$ were considered as two separate matrices. The same description can be given if we use the *combined* $m \times (p + n)$ data matrix $\mathbf{A}_{y+x}$. In this matrix the first p columns equal $\mathbf{A}_y$ and the next n columns equal $\mathbf{A}_x$. Its covariance matrix can be decomposed as:

$$\Sigma_{y+x} = \mathbf{A}_{y+x}'\mathbf{A}_{y+x} = \begin{pmatrix} \Sigma_{yy} & \Sigma_{yx} \\ \Sigma_{xy} & \Sigma_{xx} \end{pmatrix}.$$

The problem has now been reformulated as obtaining correlations between two groups of variables within the *same* data set. This formulation has been adopted in the PRAAT program.

3 A canonical correlation analysis example

As an example we will use the data set of Pols et al. (1973) which contains the first three formant frequency values and levels from the 12 Dutch monophthong vowels as spoken in /h_t/ context by 50 male speakers. This data set is available as a `TableOfReal`-object in the PRAAT program: the first three columns in the table contain the frequencies of the first three formants in Hertz and the next three columns contain the levels of the formants in decibel below the overall sound pressure level (SPL) of the measured vowel segment. There are $600 = 50 \times 12$ rows in this table. Because the levels are all given as positive numbers, a small number means a relatively high peak, a large number a relatively small peak. To get an impression of this data set we have plotted in figure 1 the logarithmically transformed and standardized first and second formant against each other. In the next subsection more details about the transformation will be given.

3.1 Finding correlations between formant frequencies and levels

We will try to find the canonical correlation between the three formant frequency values and the three levels. Instead of the frequency values in Hertz we will use logarithmic values and standardize all columns[1] (for each column separately: subtract the column average and divide by the standard deviation). Before we start the canonical correlation analysis we will first have a look at the *Pearson* correlations within this data set. This correlation matrix is displayed in the lower triangular part of table 1. In the upper triangular part are displayed the correlations for the linear frequency scale in Hertz.

[1]The standardization is, strictly speaking, not necessary because correlation coefficients are invariant under standardization.

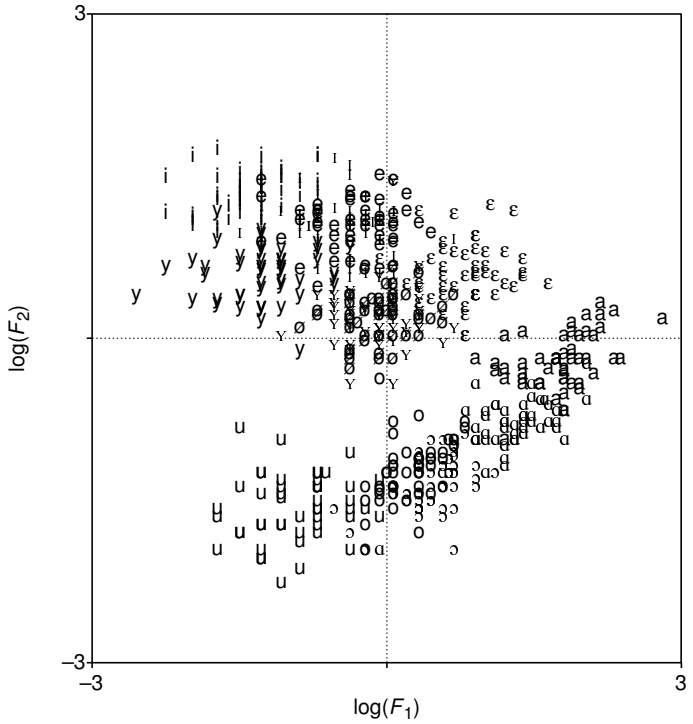

Fig. 1. The logarithmically transformed first and second formant frequencies of the Pols et al. (1973) data set.

We clearly see from the table that the correlation pattern in the upper triangular part follows the pattern from the lower triangular part for the logarithmically transformed frequencies. To get an impression of the variability of these correlations, we have displayed in table 2 the confidence intervals at a confidence level of 0.95. We used Ruben's approximation for the calculation of the confidence intervals and applied a Bonferroni correction for the significance level (Johnson, 1998). Script 1 summarizes[2].

Create TableOfReal (Pols 1973)... yes	▷ $F_{1,2,3}$ (Hz) and levels $L_{1,2,3}$.
Formula... if col < 4 then log10(self) else self endif	▷ To log($F_{1,2,3}$).
Standardize columns	
To Correlation	
Confidence intervals... 0.95 0 Ruben	▷ Bonferroni correction.

Script 1: Calculating correlations and confidence intervals.

The lower triangular part of table 1 in which the correlations of the logarithmically transformed formant frequency values are displayed, shows exact agreement with the lower part of table III in Pols et al. (1973). The correlation matrix shows that high correlations exist between some formant frequencies and some levels. According to the source-filter model of speech production vowel spectra have approximately a decli-

[2]In this script and the following ones, the essential PRAAT commands are displayed in another type family. Text that starts with a ▷-symbol is comment and not part of the script language. Note that these scripts only summarize the most important parts of the analyses. Complete scripts that reproduce all analyses, drawings and tables in this paper, can be obtained from the author's website `http://www.fon.hum.uva.nl/david/`.

Table 1. Correlation coefficients for the Pols et al. (1973) data set. The entries in the lower triangular part are the correlations for the logarithmically transformed frequency values while the entries in the upper part are the correlations for frequency values in Hertz. For better visual separability the diagonal values, which are all 1, have been left out.

	F_1	F_2	F_3	L_1	L_2	L_3
$\log(F_1)$		−0.338	0.191	0.384	−0.507	−0.014
$\log(F_2)$	−0.302		0.190	−0.106	0.530	−0.568
$\log(F_3)$	0.195	0.120		0.113	−0.036	0.019
L_1	0.370	−0.090	0.116		−0.042	0.085
L_2	−0.533	0.512	−0.044	−0.042		0.127
L_3	−0.021	−0.605	0.017	0.085	0.127	

Table 2. Confidence intervals at a 0.95 confidence level of the correlation coefficients in the lower triangular part of table 1. Confidence intervals were determined by applying Ruben's approximation and a Bonferroni correction was applied to the confidence level. The upper and lower triangular part display the upper and lower value of the confidence interval, respectively. For example, the confidence interval for the −0.533 correlation between L_2 and $\log(F_1)$ is (−0.614, −0.442).

	$\log(F_1)$	$\log(F_2)$	$\log(F_3)$	L_1	L_2	L_3
$\log(F_1)$		−0.189	0.307	0.469	−0.442	0.099
$\log(F_2)$	−0.407		0.236	0.030	0.595	−0.522
$\log(F_3)$	0.077	0.001		0.232	0.076	0.136
L_1	0.262	−0.207	−0.004		0.078	0.203
L_2	−0.614	0.417	−0.162	−0.161		0.243
L_3	−0.140	−0.675	−0.103	−0.035	0.007	

nation of −6 dB/octave which indicates that a strong linear correlation between the logarithm of the formant frequency and the formant level in decibel should exist.

To obtain the canonical correlations between the formant frequencies and formant levels we first let the PRAAT program construct a `CCA`-object from the `TableOfReal`-object. This object will next be queried for the canonical correlations. In the construction of the `CCA`-object, the first three columns in the `TableOfReal`-object, those that contain the formant frequencies, are associated with the matrix $\mathbf{A}_y$, and the last three columns that contain the formant levels are associated with the matrix $\mathbf{A}_x$. Then, the calculations as outlined in section 2.2.2 are used to determine the canonical correlations. Script 2 summarizes.

```
select TableOfReal pols_50males          ▷ The log(F) values.
To CCA... 3                              ▷ We have 3 dependent variables.
Get correlation... 1
Get correlation... 2
Get correlation... 3
```

Script 2: Canonical correlation analysis.

In table 3 we show the canonical correlations together with the eigenvector loadings on the variables. The eigenvectors belonging to the first and the second canonical correlation have also been drawn in figure 2 with a continous line and a dotted line,

Table 3. The canonical correlations between formant frequencies and formant levels and their corresponding eigenvectors.

	ρ	$\log(F_1)$	$\log(F_2)$	$\log(F_3)$	L_1	L_2	L_3
1	0.867	−0.187	0.971	−0.148	−0.092	0.714	−0.694
2	0.545	0.891	0.443	−0.099	0.646	−0.428	−0.632
3	0.072	0.166	0.017	−0.986	−0.788	−0.530	−0.313

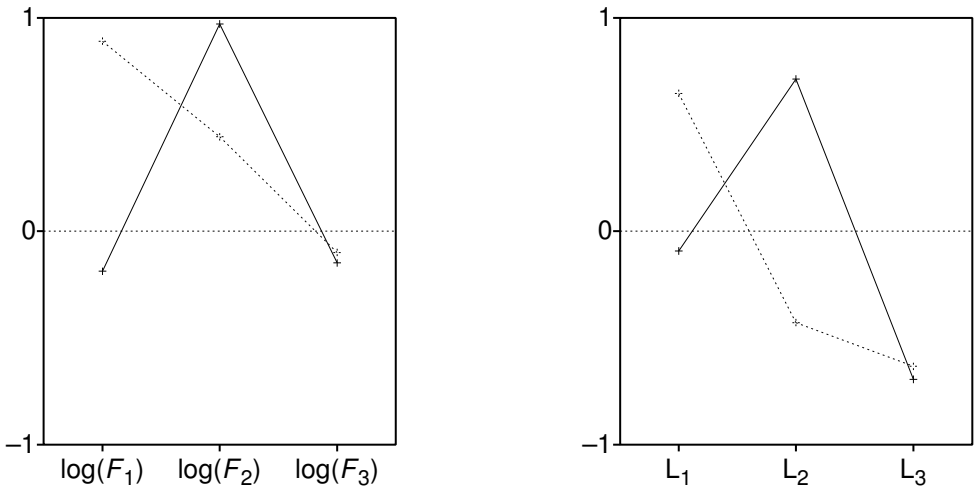

Fig. 2. The eigenvectors corresponding to the first (continuous line) and the second canonical correlation(dotted line).

respectively. In this figure the plot on the left shows the weighting of the frequencies. We see that, for the first eigenvector, most of the weight is put on $\log(F_2)$, and that the other two frequencies are barely weighted. On the other hand, for the weighting of the levels, the first eigenvector shows approximately equal weighting of the second and third level (in absolute sense). This is confirmed by the data in table 1 that show a high correlation, 0.512, between $\log(F_2)$ and L_2 and the highest correlation, 0.605, between $\log(F_2)$ and L_3. Table 3 indicates that the weightings of L_2 and L_3 in the first eigenvector are even larger than in table 1.

3.2 Using the correlations for prediction

The outcome of the canonical correlation analysis on the Pols et al. data set was three canonical correlations, ρ_i, with their associated eigenvectors $\mathbf{x}_i$ and $\mathbf{y}_i$. These eigenvectors can be used to construct the scores (canonical variates) $\mathbf{z}_y$ and $\mathbf{z}_x$ as was shown in equations (7) and (6), respectively. In figure 3 we have drawn a scatter plot of the first canonical variates. The straight line shows the empirical relation $y_1 = 0.867x_1$ for the first canonical correlation. We note two separate clusters, one for the back vowels and another for the front vowels. The main ordering principle in the figure is from front to back, as can also be seen from the first eigenvector for the formants in figure 2 which is dominated by the second formant frequency. The linear part of the relation between

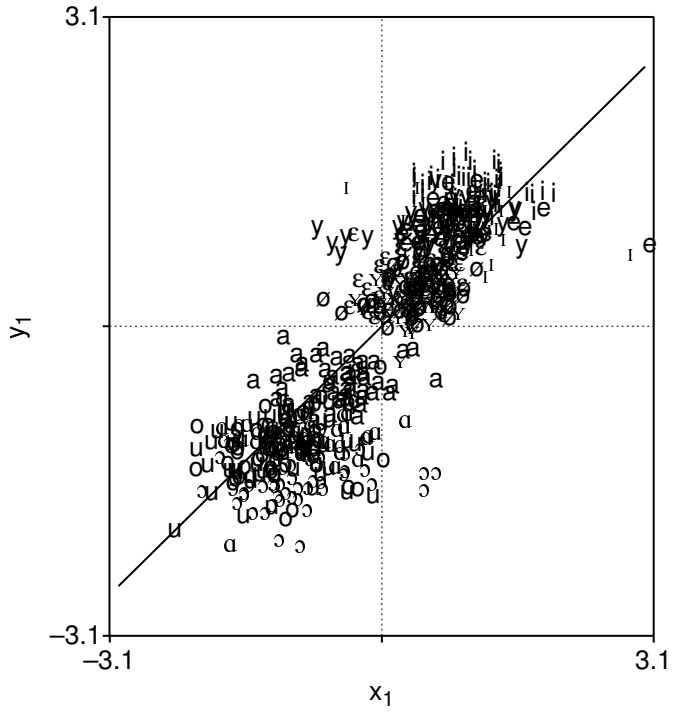

Fig. 3. A scatter plot of the first canonical variates. The straight line shows the canonical correlation relation $y_1 = \rho_1 x_1$, where $\rho_1 = 0.867$.

these canonical variables can be exploited by predicting one from the other. In the following we will try to predict formant frequency values from formant levels. We start with the equations for the canonical variates and write:

$$\mathbf{z}_{y,i} = \rho_i \mathbf{z}_{x,i}, \text{ for } i = 1, 2 \text{ and } 3. \tag{31}$$

These three equations show the optimal linear relation between a linear combination of formant frequencies and formant levels, the $\mathbf{z}_y$ and the $\mathbf{z}_x$, respectively. Equation (31) could also be interpreted as a prescription to determine the $\mathbf{z}_y$'s when only the $\mathbf{z}_x$'s are given. In the equation above the vectors $\mathbf{z}$ are m-dimensional. For every element j of the vectors $\mathbf{z}$, we can substitute back the original variables and obtain the following equation

$$\mathbf{Yf} = \mathbf{D}(\rho_i)\mathbf{Xl}, \tag{32}$$

where $\mathbf{f}$ and $\mathbf{l}$ are the vectors with the three formant frequencies and levels, respectively, and $\mathbf{D}(\rho_i)$ is a diagonal matrix. The $\mathbf{Y}$ and $\mathbf{X}$ are the eigenvectors. Now, because the $\mathbf{Y}$ are orthogonal, we can write the solution as

$$\mathbf{f} = \mathbf{Y}'\mathbf{D}(\rho_i)\mathbf{Xl}. \tag{33}$$

When we train a discriminant classifier with the standardized formant frequency values and use the same set we used for the training as input for the classifier we obtain 73.9% correct classification with the 12-label set (discriminant analysis with the PRAAT program has been covered in Weenink (1999)). When we use the formant *levels* to predict the formant *frequencies* and subsequently use these predicted formant frequencies as input to the classifier, we get 26% correct classification. Using only formant levels for discriminant classification gives 37.2% correct. Both classifications are above chance (8.5%). The following script summarizes.

```
select TableOfReal pols_50males
plus CCA pols_50males
Predict... 4                              ▷ Start column is 4.
Select columns where row... "1 2 3" 1     ▷ Select only F1, F2, F3.
Rename... f123
To Discriminant                           ▷ Train the classifier.
plus TableOfReal f123
To ClassificationTable... y y             ▷ Use linear discriminant.
To Confusion                              ▷ Get the confusion matrix.
fc = Get fraction correct
```

Script 3: Prediction from canonical correlations.

4 Principal components and auto-associative neural nets

4.1 Introduction

In this section we try to use canonical correlation analysis to demonstrate that appropriately chosen neural nets can also perform principal component analysis. We will do so by comparing the output from an auto-associative neural net with the output of a principal component analysis by means of canonical correlation analysis. As test data set we will use only the three formant frequency values from the Pols et al. data set. In order to make the demonstration not completely trivial we compare two-dimensional representations. This means that in both cases some data reduction must take place.

4.2 The auto-associative neural net

An auto-associative neural net is a supervised neural net where each input is mapped to itself. We will use here the supervised feedforward neural net as is implemented in the PRAAT program. Auto-associativity in these nets can best be accomplished by making the output units linear[3] and the number of dimensions of the input and output layer must be equal too (Weenink, 1991). The trivial auto-associative net has no hidden layers and maps its input straight to its output. Interesting things happen when we compress the input data by forcing them through a hidden layer with less units than the input layer. In this way the neural net has to learn some form of data reduction. This reduction probably must be some way of principal component analysis in order to maintain as much variation as possible in the transformation from input layer to output layer.

Since our input data is three-dimensional, the number of input and output nodes for the neural network is already fixed and the only freedom in the topology that is left is the number of hidden layers and the number of nodes in each hidden layer. To keep the comparison as simple as possible, we will use only one hidden layer in this task with two nodes in this layer. The resulting topology for the supervised feedforward neural net is a (3,2,3) topology, i.e., 3 input nodes, 2 hidden nodes and 3 output nodes. A network with this topology has only 17 adaptable weight: 9 weights for the output layer and 8 weights for the hidden layer. The topology of this network is displayed in figure 4.

In the training phase we try to adjust the weights of the network in such a way that when we propagate an input through the neural net, the output activation of the neural net will equal the input. Of course this is not always possible for all inputs and therefor we try to make them as close as possible on the average. Closeness is then

[3]This linearity is only for the output nodes, the hidden nodes still have the sigmoid non-linearity.

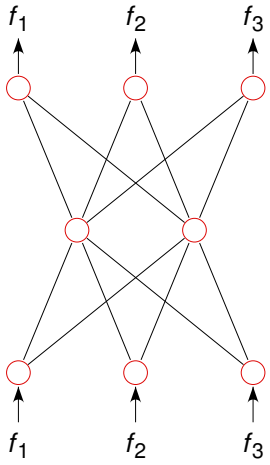

Fig. 4. Topology of the supervised auto-associative feedforward neural net used for learning the associations between logarithmically scaled formant frequency values.

mathematically defined as a minimum squared error criterion.

4.3 Data preprocessing

In order to guarantee proper training we have to arrange for all inputs to be in the interval (0, 1). We have scaled all formant frequency values as

$$f_i = \log \frac{F_i}{(2i-1)500} + 0.5, \text{ for } i = 1, 2 \text{ and } 3. \tag{34}$$

In this formula formant frequencies F_i in Hertz are first scaled with respect to the resonance frequencies of a straight tube which are at frequencies of $(2i-1)500$ Hz. Next the logarithm of this fraction is taken[4]. Since the logarithm of this fraction can take on negative values we add the factor 0.5 to make the number positive.

To show the effect of this scaling we have drawn in figure 5 the box plots of the data before and after the scaling. A ”box plot”, or more descriptively a ”box-and-whiskers plot”, provides a graphical summary of data. The box is marked by three continuous horizontal lines which, from bottom to top, indicate the position of the first, second and third quartile. The box height therefor covers 50% of the data (the line of the second quartile shows of course the position of the *median*). In the PRAAT version of the box plot, the box has been extended with a dotted line that marks the position of the average. The lengths of the vertical lines, the ”whiskers”, show the largest/smallest observation that falls within 1.5 times the box height from the nearest horizontal line of the box. If any observations fall farther away, the additional points are considered ”extreme” values and are shown separately.

[4]It is not strictly necessary to take the logarithm. The scaling with the corresponding uneven multiple of 500 Hz for each formant is already sufficient to render all values in the interval (0.4,2.2]. Subsequently dividing by a factor somewhat greater than 2.2 would yield numbers in the (0,1) interval. Taking an extra logarithm, however, achieves a somewhat better clustering. A discriminant classification with equal train set and test set shows 73.9% correct for the logarithmic scaling, as was already shown in section 3.2, versus 72.8% for the alternative scaling discussed in this footnote.

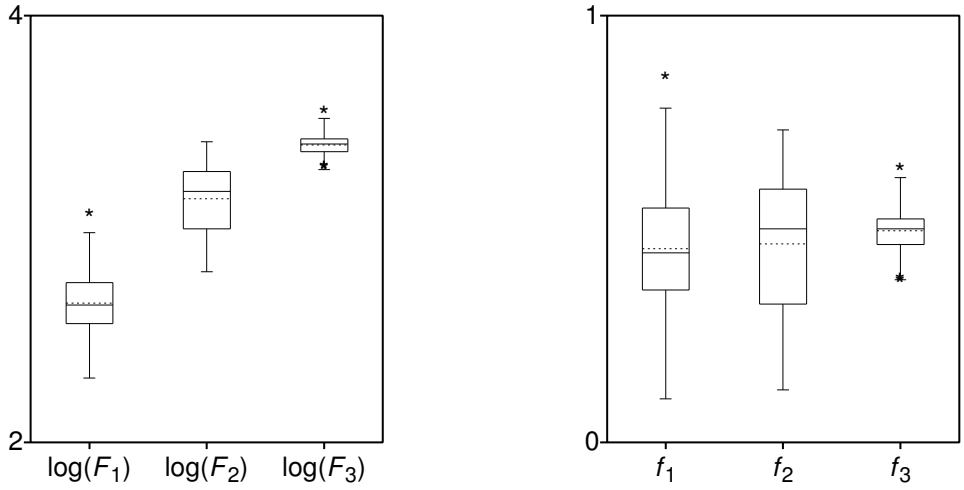

Fig. 5. Box plots before (left) and after (right) scaling the logarithmically transformed frequency values. The f_i are scaled to the interval (0, 1) according to equation (34). The dotted lines in the box plots indicate the average values.

Besides scaling the values to the (0, 1) interval we also note that the locations of the scaled formant frequency values have become more equalized. The following script summarizes the scaling.

```
Create TableOfReal (Pols 1973)... no                ▷ Only frequencies, no levels.
Formula... log10 (self / ((2*col-1)*500) + 0.5      ▷ Equation (34).
```

Script 4: Scaling of the formant frequencies to the (0, 1) interval.

4.4 Training the neural net

After processing the data we finally have a table in which all elements are within the (0, 1) interval. We duplicate this table and cast the two resulting objects to a `Pattern`-object and an `Activation`-object, respectively. These two objects function as the input and output for the auto-associative feedforward neural net. The next step is then to create a neural net of the right topology, select the input and the output objects and start learning. Preliminary testing showed that 500 learning epochs were sufficient for learning these input-output relations.

Because the learning process uses a minimization algorithm that starts the minimization with random weights, there always is the possibility to get stuck in a local minimum. We can not avoid these local minima. However, by repeating the minimization process a large number of times, each time with different random initial weights, we can hope to find acceptable learning in some of these trials. We therefor repeated the learning process 1000 times and each time used different random initial weights. The repeated learning only took 27 minutes of cpu-time on a computer with a 500 MHz processor. It turned out that after these 1000 learning sessions all the obtained minima were very close to each other. The distribution of the minima in this collection of 1000 was such that the absolute minimum was 0.5572, the 50% point (median) was at 0.5575 and the 90% point at 0.5580. If we consider that the training set had 600 records and

each record is a 3-dimensional vector with values in the interval (0, 1) and this minimum is the sum of all the squared errors then excellent learning has taken place. We have stored the weights of the neural net that obtained the lowest minimum. Script 5 summarizes the learning process.

```
min_global= 1e30                                          ▷ Initialize to some large value.
Create Feedforward Net... 3_2_3 3 3 2 0 y                 ▷ Topology (3, 2, 3).
for i to 1000
    select FFNet 3_2_3
    Reset... 0.1                                          ▷ All weights random uniform in [-0.1, 0.1].
    plus Activation pols_50males
    plus Pattern pols_50males
    Learn (SM)... 500 1e-10 minimum squared error         ▷ 500 epochs.
    select FFNet 3_2_3
    min = Get minimum
    if min < min_global
        min_global = min
        Write to short text file... 3_2_3                 ▷ Save FFNet-object to disk.
    endif
endfor
```

Script 5: Training the neural net.

4.5 The comparison

Now that the best association between the three-dimensional outputs and inputs by means of two hidden nodes has been learned by the neural net, we want to compare this mapping with the results of a two-dimensional principal component analysis. We want to obtain the representation of all the inputs at the two nodes of the hidden layer. This can be done by presenting an input to the trained neural net, let the input propagate to the first hidden layer and then record the activation of the nodes in this layer. The input to the neural net will therefor be a 600×3 table and the output will be the activation at the hidden layer, a table of dimension 600×2. Script 6 summarizes.

```
select FFNet FFNetmin                                     ▷ Select the trained neural net
plus Pattern pols_50males                                 ▷ + the input.
To Activation... 1                                        ▷ Layer 1 is the hidden layer.
```

Script 6: Get activation at hidden layer.

The mapping to the principal component plane of the scaled data is simple to obtain. See for example Weenink (1999) for more information on principal component analysis. The first two principal components explain 95.8% of the variance. Script 7 summarizes.

To get more insight in the results of the two different analysis we have plotted in figure 6 the neural net and principal component representations of the formant data preprocessed according to equation (34). The figure on the left shows the representation in the hidden layer, the figure on the right displays the data in the principal component plane. Both representations look very similar and closer inspection shows that they are almost reflected versions of each other. When we compare them to figure 1, we notice a great resemblance which shows that predominantly only the first two formant frequencies contribute to the representations in figure 6.

```
Create TableOfReal (Pols 1973)... no                    ▷ No levels.
Formula... log10 (self / ((2*col-1)*500) + 0.5
To PCA                                                  ▷ Principal Component Analysis.
vaf = Get fraction variance accounted for... 1 2
plus TableOfReal pols_50males
To Configuration... 2                                   ▷ The 2-dimensional mapping.
```

Script 7: Mapping to the principal component plane.

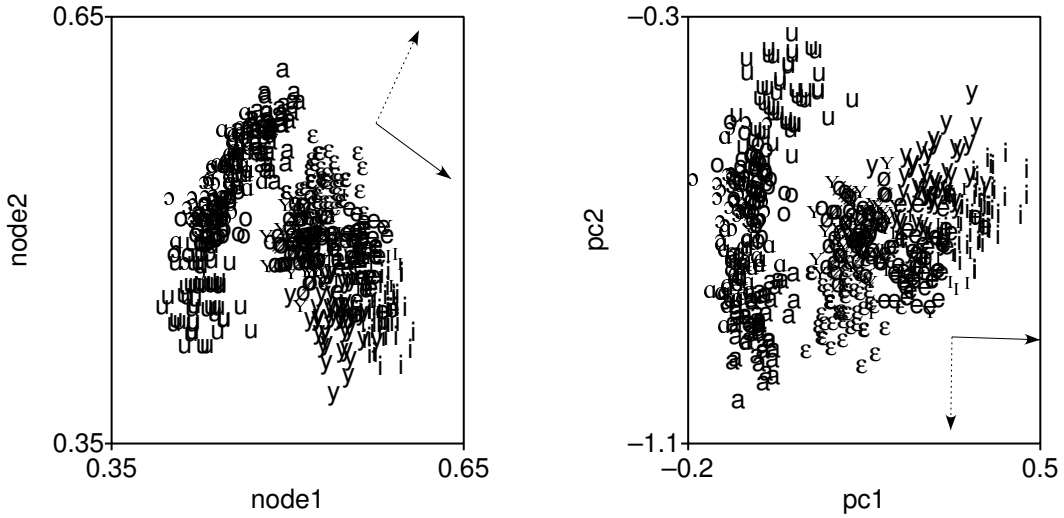

Fig. 6. Two different representations of the formant frequency data scaled according to equation (34). Left: the representation at the hidden layer of the neural net of figure 1 with topology (3, 2, 3). Right: the principal components plane of the first two principal components. The plain and dotted arrows are data taken from table 5 and indicate the directions of the eigenvectors for the first and second canonical correlation, respectively.

We can now combine the two representations in one 600 × 4 data matrix and calculate the correlations between the columns of this matrix. The correlation coefficients are shown in the upper diagonal part in table 4. The following script summarizes.

```
select TableOfReal hidden
plus TableOfReal pca
Append columns                                          ▷ Now 2 times 2 columns → 4 columns.
Rename... hidden_pca
To Correlation
```

Script 8: Correlations between the hidden layer and the principal component representations.

For the principal components, the table confirms that the correlation coefficient between the first and the second principal component is zero, as it must be of course, since the whole purpose of principal component analysis is removing correlations between dimensions. The representations at the two hidden nodes are not independent as the (negative) correlation coefficient between node 1 and node 2 shows. Substantial correlations exist between the two neural dimensions and the principal component dimensions. However, the two plots in figure 6 suggest that there is more correlation than is shown in the table. This is where a canonical correlation analysis can be useful. The results of the canonical correlation analysis between the two-dimensional

Table 4. The correlation coefficients for the combined representations of formant frequencies at the hidden nodes of a neural network and principal components. The lower diagonal part contains the correlations after a Procrustus similarity transform on the hidden nodes representation. For clarity, diagonal ones have been left out.

	node1	node2	pc1	pc2
node1′		−0.363	0.927	−0.376
node2′	−0.055		−0.686	−0.727
pc1	1.000	−0.029		0.000
pc2	−0.025	1.000	0.000	

Table 5. Characteristics of the canonical correlation analysis between the two-dimensional representation of formant frequencies at the hidden nodes of a neural network and the two principal components. Canonical correlation coefficients and corresponding pairs of eigenvectors are shown.

	ρ	node1	node2	pc1	pc2
1	1.000	0.854	−0.520	0.999	−0.033
2	1.000	0.488	0.873	−0.017	−1.000

representation at the hidden nodes and the two-dimensional principal component representation are displayed in table 5. Besides canonical correlation coefficients, the table also shows the eigenvectors. Additionally, the eigenvectors are graphically displayed in figure 6 with arrows. The two arrows in the left and the right plot, drawn with a plain line, are the directions of maximum correlation between the two representations: when we project the 600 two-dimensional data points on these directions, the resulting two 600-dimensional data vectors have the maximum obtainable canonical correlation coefficient of 1.000. The second coefficient also equals 1, rounded to three digits of precision. The corresponding eigenvectors are drawn as the arrows with a dotted line. In figure 7 we have plotted the canonical variates (scores) for this analysis. Script 9 summarizes.

```
select TableOfReal hidden_pca                    ▷ 4 columns.
To CCA... 2                                      ▷ 2 dependent variables.
plus TableOfReal hidden_pca
To TableOfReal (scores)... 2
```

Script 9: Get canonical variates (scores).

We see from the plots in figure 7 a nice agreement between the scatter plots of the neural net scores on the left and the principal component scores on the right. However, we note from figure 6 that the two eigenvectors **y** are not mutually orthogonal. The same occurs for the two eigenvectors **x**, they are not orthogonal either (although harder to see in the figure, the numbers in table 5 will convince you). This is a characteristic of equations like (4) and (5): in general these equations don't have eigenvectors that are orthogonal. Because the scores (canonical variates) are obtained by a projection of the original data set on the eigenvectors of the canonical correlation analysis, the resulting scatter plots will show a somewhat distorted map of the original data. This is in contrast with principal component analysis where the eigenvectors are orthogonal and therefor

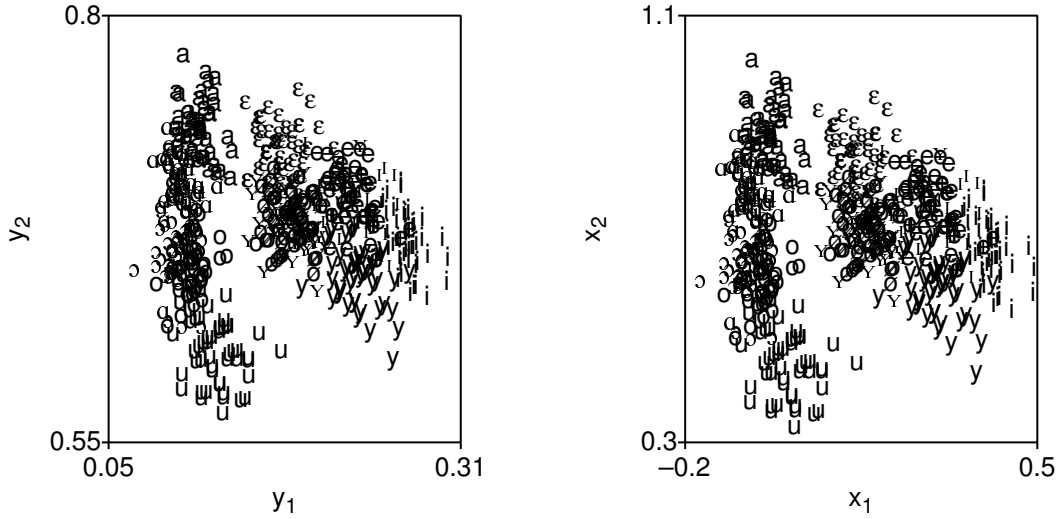

Fig. 7. Scatter plots of canonical variates for the dependent (left) and the independent data set (right). The dependent and independent data sets are the neural net data and the principal component data set, respectively.

the new principal dimensions are a mere rotation of the original dimensions. This means that a principal component analysis does not change the structure of the data set and relative distances between the points in the data set are preserved. In the mapping to the canonical variate space, the structure of the data set is not preserved and the relative distances have changed.

4.6 Procrustus transform

It is possible, however, to transform one data set to match another data set, as closely as possible, in which the structure of the transformed data set is preserved. This similarity transformation is called a Procrustus transform. In the transform the only admissible operations on a data set are dilation, translation, rotation and reflection and we can write the equation that governs the transformation of data set **X** into **Y** as follows:

$$\mathbf{Y} = s\mathbf{X}\mathbf{T} + \mathbf{1}\mathbf{t}'. \tag{35}$$

In this equation s is the dilation or scale factor, **T** is an orthogonal matrix that incorporates both rotation and reflection, $\mathbf{t}'$ is the translation vector, and **1** is a vector of ones. Given data sets **X** and **Y**, a Procrustus analysis delivers the parameters for s, **t** and **T**. The equation above transforms **X** into **Y**. The inverse, the one that transforms **Y** into **X** can easily be deduced from equation (35) and is:

$$\mathbf{X} = \frac{1}{s}(\mathbf{Y} - \mathbf{1}\mathbf{t}')\mathbf{T}'. \tag{36}$$

More details of the Procrustus transform and the analysis can be found in Borg & Groenen (1997). In figure 8 we show the result of a Procrustus analysis on the neural net and the principal component data sets. The plot on the left is the Procrustus transform of the neural net data set and was obtained from the plot in figure 6 by a clockwise rotation with an angle of approximately 31°, followed by a reflection around the horizontal axis,

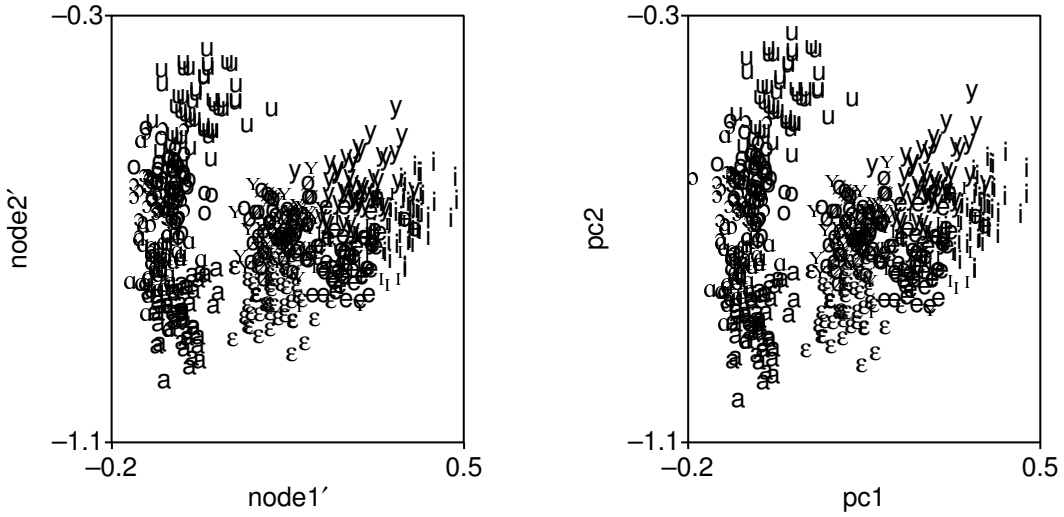

Fig. 8. Scatter plots of the Procrustus-transformed neural net representation (left) and the principal component representation (right). The plot on the left is obtained from the left plot in figure 6 by a clockwise rotation of 31°, followed by a reflection around the horizontal axis, a scaling by a factor 2.98 and a translation with the vector (-0.42, 1.35). The plot on the right is only for comparison and shows the same data as the plot on the right in figure 6.

a scaling by a factor 2.98 and a translation with the vector (-0.42, 1.35). The parameters for this transform were obtained from matching the two-dimensional neural net data set with the two-dimensional principal component data set. The two plots now look very similar. In table 4 we show in the lower diagonal the correlation coefficients between the Procrustus-transformed neural net data set and the principal component data set. These correlations were also obtained, in a manner analogous to the data in the upper diagonal part, by appending columns into a combined data set. Script 10 summarizes.

```
select Configuration pca
plus Configuration hidden
To Procrustus
plus Configuration hidden
To Configuration                          ▷ Apply Procrustus.
Rename... hiddenp
To TableOfReal
plus TableOfReal pca
Append columns                            ▷ Combine the two tables.
To Correlation
```

Script 10: Correlation of Procrustus-transformed data with principal components.

When we compare corresponding data elements above and below the diagonal in table 4, we notice that node1′ and node2′ have become more decorrelated as compared to node1 and node2, making these new dimensions more independent from each other. The pc1 and pc2 have not changed and therefor remain uncorrelated. And, finally, the correlations between node1′ and pc1 and, especially, between node2′ and pc2 have increased and are almost perfect now.

4.7 Summary

All the data presentations in the preceding sections have shown that there is a great amount of similarity between the internal representation of a auto-associative neural net and a principal component analysis for the Pols et al. formant frequency data set. Although the presentation in these sections present no formal proof and were only used as a demonstration of some of the methods available in the PRAAT program, we hope that it has been made plausible that auto-associative neural nets and principal components bear a lot in common.

5 Discussion

We have shown that the canonical correlation analysis can be a useful tool for investigating relationships between two representations of the same objects. Although the mathematical description of the analysis that has been given in this paper can be considered as a *classical* analysis, the results can also be used with modern robust statistics and data reduction techniques. These modern techniques are more robust against outliers. Essential to these modern techniques is a robust determination of the covariance matrix and the associated mean values (Dehon et al., 2000). The description we have given in section 2.2.1 does not prescribe how a covariance matrix is obtained and could therefor be used with these modern techniques.

Acknowledgement

The author wants to thank Louis Pols for his critical review and constructive comments during this study.

References

Boersma, P. P. G. & D. J. M. Weenink (1996): *Praat, a system for doing phonetics by computer, version 3.4*, Report 132, Institute Of Phonetic Sciences University of Amsterdam (for an up-to-date version of the manual see `http://www.fon.hum.uva.nl/praat/`).

Borg, I. & P. Groenen (1997): *Modern Multidimensional Scaling: Theory and Applications*, Springer Series in Statistics, Springer.

Dehon, C., P. Filzmoser & C. Croux (2000): *Robust methods for canonical correlation analysis*, pp. 321–326, Springer-Verlag, Berlin.

Golub, G. H. & C. F. van Loan (1996): *Matrix Computations*, The John Hopkins University Press, 3rd edn.

Hotelling, H. (1936): "Relations between two sets of variates", *Biometrika* **28**: pp. 321–377.

Johnson, D. E. (1998): *Applied Multivariate Methods for Data Analysts*, Duxburry Press.

Pols, L. C. W., H. Tromp & R. Plomp (1973): "Frequency analysis of Dutch vowels from 50 male speakers", *J. Acoust. Soc. Am.* **53**: pp. 1093–1101.

Weenink, D. J. M. (1991): "Aspects of neural nets", *Proceedings of the Institute of Phonetic Sciences University of Amsterdam* **15**: pp. 1–25.

Weenink, D. J. M. (1999): "Accurate algorithms for performing principal component analysis and discriminant analysis", *Proceedings of the Institute of Phonetic Sciences University of Amsterdam* **23**: pp. 77–89.