Lab Communications Project 7

Correlation, Coherence, and Information Flow

1. Preface

The purpose of this project is to familiarize you with the concepts of correlation, coherence and information flow. Correlation describes the similarity between two signals, whereas coherence measures the dependency of two signals with respect to the frequency. Information flow shows the impact of one signal to another signal.

2. Correlation

Correlation refers to the statistical relationship between two random variables. Cross-correlation is the measure of similarity between two signals, when one of them is subjected to time lag. The correlation coefficient is a measure of linear dependence between two signals, giving values between $+1$ and -1. A positive value indicates that two signals are positively correlated, zero refers to no correlation while a negative value expresses negative correlation. The autocorrelation is the cross-correlation of a signal with its own time lagged version. Cross-correlation between two time series $x(t)$ and $y(t)$ can be calculated by

$$R_{xy}(t) = \sum_{\tau=-\infty}^{+\infty} x(\tau - t) y(\tau). \tag{1}$$

The correlation coefficient between two time series is a scalar value, showing the strength of the linear relationship between them. It can be calculated by

$$r_{xy} = \frac{s_{xy}}{\sqrt{s_{xx}}\sqrt{s_{yy}}} \tag{2}$$

for the two signals $x(t)$ and $y(t)$. The sample variance can be calculated by using

$$s_{xy} = \frac{1}{N}\sum_{j=1}^{N}(x_j - \bar{x})(y_j - \bar{y}) \tag{3}$$

with the sample means $\bar{x}$ and $\bar{y}$. They can be calculated by (shown for x, y analogously)

$$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x(n)\,. \tag{4}$$

2.1. Exercise

(a) Generate the vectors $\mathtt{x} = [9\ 2\ 6\ 5\ 8]$ and $\mathtt{y} = [12\ 8\ 6\ 4\ 10]$.

(b) Calculate the sample mean $\bar{x}$ and $\bar{y}$ for both vectors by using expression (4).

(c) Calculate the sample variance s_{xx} and s_{yy} for both vectors using expression (3). Additionally, calculate the sample covariance s_{xy}.

(d) Calculate the sample correlation coefficients r_{xx}, r_{yy}, and r_{xy} using expression (2).

(e) Compare your results of the correlation coefficient with the output of the command given below:
`corrcoef(x,y)`.

(f) Generate a vector `z = [1 3 -1 2]`.

(g) Calculate the autocorrelation vector of `z` from delay $t = -3$ to 0 using expression (1).

(h) Correlation is similar to convolution, with the only difference being a time reversal. We know that time reversal in time domain is equal to taking the complex conjugate in frequency domain. This means that we can also calculate the autocorrelation of a signal using its Fourier transform. Calculate the autocorrelation of the vector given in (f) using the expression:

$$R_{zz}(n) = \text{IFFT}\left[\text{FFT}\left\{z(n)\right\} \cdot \text{FFT}\left\{z(n)\right\}^*\right]. \tag{5}$$

If you use the function `fft`, please also specify a reasonable FFT-size when calling the function.

(i) Compare the results of parts (g) and (h).

3. Coherence

Coherence is used to identify those variations between two signals which have similar spectral properties. Coherence can be described by the expression

$$\text{Coh}(\omega) = \frac{\mid S_{xy}(\omega) \mid^2}{S_{xx}(\omega) S_{yy}(\omega)} \tag{6}$$

where $S_{xy}(\omega)$ is the cross power spectral density between the signals $x(n)$ and $y(n)$, while $S_{xx}(\omega)$ and $S_{yy}(\omega)$ are the auto power spectral densities of the signals $x(n)$ and $y(n)$, respectively. The cross power spectral density between $x(n)$ and $y(n)$ can be calculated as:

$$S_{xy}(\omega) = \text{FFT}\left\{x(n)\right\} \cdot \text{FFT}\left\{y(n)\right\}^*. \tag{7}$$

In this exercise we will also use Welch's method to estimate power spectral densities.

3.1. Exercise

(a) Generate two vectors $x(n)$ and $y(n)$ using expression given below in MATLAB

```
fs = 1000;t = 0:1/fs:1-1/fs;
x = sin(2*pi*200*t)+0.5*randn(size(t));
y = 0.35*cos(2*pi*200*t)+0.5*randn(size(t));
```

(b) Use the MATLAB functions `cpsd` and `pwelch` to calculate the cross power spectral density between the two vectors `x` and `y` and their auto power spectral density. Description about these functions can be read by typing `help <function name>`. Please use no window (`window=[]`), no overlap (`noverlap=[]` and an FFT-size appropriate for the signal length.

(c) Finally, use expression (6) to calculate and plot the coherence. Which frequency is significantly pronounced?

(d) Use the MATLAB function `mscohere` to plot the coherence and compare your results of tasks (c) and (d).

(e) Create another random signal $p(n)$ with a length of 524288 using `randn`.

(f) Filter the signal $p(n)$ with a filter with the transfer function

$$Y(z) = P(z)(z^2 + 0.5z^5) \tag{8}$$

by using the function `filter`. Store the result in the variable `y1`.

(g) Create another random signal `y2` of the same length as that of `p`.

(h) Now, calculate the coherence between `p` and `y1` by using expression (6) and (7) and compare it with the coherence between `p` and `y2`. Use the complete signal for this calculation.

(i) Now divide all signals into windows of length 1024 and calculate the PSD for each window again using (7). Take the average of all these windows and compute the coherence using (6). Finally compare your results with (h).

4. Information flow

Information flow or causality is the relation between two events, where one of them is understood as a consequence of the other event. In signal processing we use the concept of information flow to determine in which order events started. This can be better illustrated by the following expressions:

$$x_1(t) = a_{11}x_1(t-1) + a_{12}x_2(t-1) + n_1(t), \tag{9}$$

$$x_2(t) = a_{21}x_1(t-1) + a_{22}x_2(t-1) + n_2(t). \tag{10}$$

Both time series are modelled as autoregressive processes with order 1, AR(1). The coefficients of causality a_{xx} and their magnitudes show the extent of the information flow between the time series.

For a better understanding of the following exercise, please read the appendix to this lab documentation before you start with this exercise.

4.1. Exercise

(a) Generate two time series of length 1000 satisfying (9) and (10), using the following AR coefficients and a random initialization of the samples $x_1(1)$ and $x_2(1)$ (use `randn`).
$a_{11} = 0.8$
$a_{12} = 0.2$
$a_{21} = 0.6$
$a_{22} = 0.2$

(b) Add appropriate random noise to these two time series by adding the noise terms $n_1(t)$ and $n_2(t)$ as given in equations 9 and 10.

(c) Construct a 999-by-5 zero matrix `K`. Replace its first column with a vector of ones (999-by-1). The second and third column should be replaced by data vectors (first 999 data points). The fourth and fifth column should also be replaced by data vectors (last 999 data points).

(d) Subject this matrix `K` to orthogonal triangular decomposition using the MATLAB function `qr` and store the output in variable `R`. The description of this function can be found by typing `help qr`.

(e) Extract the upper triangular part of the variable `R` using the MATLAB function `triu` and store the output in the same variable. The description of this function can be found by typing `help triu`.

(f) Extract the first three rows of the first three columns of `R` and store them in the variable `R11`.

(g) Extract the first three rows of the fourth and fifth column of `R` and store them in the variable `R12`.

(h) Use the backslash-operator to solve the linear equation system (`R11\R12`), take its transpose and store the result in the variable `A`.

(i) The second and third column of `A` contain the estimated AR coefficients. Compare these results with the actual values.

Appendix

A. Disclaimer

This appendix is currently work in progress and may contain mistakes. However, hopefully it will help you write the report and understand the contents of the last part of the lab.

B. General idea of the exercise

The purpose of this exercise is to first generate two time series (signals) of length 1000 that are generated by using a (vector) autoregression model ((V)AR-model) with given coefficients a_{ij} with a random initialization for the signals at $n = 0$. These observations are then disturbed by additive noise. Afterwards the goal is to extract the AR coefficients out of the disturbed observations as it would be the case in a real-world application.

C. Fast solution (not part of the lab)

Given the model order of the underlying AR-process and the observations of the two time series a linear equation system can be set up which has to be solved in order to extract the desired AR-coefficients. For the given problem the model order is $p = 1$. If we consider the equations

$$x_1(n) = a_{11}x_1(n-1) + a_{12}x_2(n-1) + d_1(n) \tag{11}$$

and

$$x_2(n) = a_{21}x_1(n-1) + a_{22}x_2(n-1) + d_2(n) \tag{12}$$

and combine them into matrix notation we get

$$\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{B} + \boldsymbol{D} \tag{13}$$

with $\boldsymbol{X}$ being the stacked vectors $x_1(n)$ and $x_2(n)$ (dim. `N x 2`), $\boldsymbol{Z}$ the stacked vectors of old signal samples $x_1(n-1)$ and $x_2(n-1)$ (dim. `N x 2`), $\boldsymbol{B}$ the matrix containing the AR-coefficients (dim. `2 x 2`), and $\boldsymbol{D}$ (dim. `N x 2`) the stacked additive noise samples $d_1(n)$ and $d_2(n)$. N is called the signal length.

For clarification, the matrix $\boldsymbol{B}$ looks like this:

$$\boldsymbol{B} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{\mathrm{T}} \tag{14}$$

If we use all but the last observations for the matrix $\boldsymbol{X}$ and all but the first observations for the matrix $\boldsymbol{Z}$ and additionally neglect the noise terms contained in the matrix $\boldsymbol{D}$ the AR-coefficients can be estimated by solving the linear equation system formed by

$$\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{B}. \tag{15}$$

In order to do so we make use of the *backslash operator* in `MATLAB`. To obtain the desired solution we enter `B=X\Z` and thus output and store the AR-coefficients.

D. Detailed solution (part of the lab)

In the lab we solve the problem in a more detailed fashion. Nevertheless, we also solve the linear equation system as given in equation (15). The first difference is the underlying AR-model and so we use the following equations instead of the ones mentioned previously:

$$x_1(n) = c_1 + a_{11}x_1(n-1) + a_{12}x_2(n-1) + d_1(n) \tag{16}$$

and

$$x_2(n) = c_2 + a_{21}x_1(n-1) + a_{22}x_2(n-1) + d_2(n) \tag{17}$$

The linear equation system stays the same but the matrix $\boldsymbol{Z}$ contains a column of ones (first column) additionaly now and the matrix $\boldsymbol{B}$ contains a row with the constants c_1 and c_2. By these changes the differences in the AR-models are completely respected.

A very well known method to solve linear equation systems is the Gaussian elimination. The basic idea is to combine the values for the past samples and the current samples (input and output) into one matrix and transform it to an upper triangular matrix for the input part of the matrix and afterwards solve the equation system by back-substitution (Assuming the equation system is sufficiently determined).

In the lab we follow this idea by combining the matrices $\boldsymbol{Z}$ and $\boldsymbol{X}$ into the matrix

$$\boldsymbol{K} = \begin{bmatrix}\boldsymbol{Z} & \boldsymbol{X}\end{bmatrix} = \begin{bmatrix} 1 & x_1(n-1) & x_2(n-1) & x_1(n) & x_2(n) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{18}$$

In order to transform it to an upper triangular matrix we use the QR decomposition which yields an upper triangular matrix $\boldsymbol{R}$ and an orthogonal matrix $\boldsymbol{Q}$:

$$\boldsymbol{K} = \boldsymbol{Q}\boldsymbol{R}. \tag{19}$$

The triangular matrix $\boldsymbol{R}$ can now be used to calculate the AR-coefficients by hand by back-substitution. However, due to time constraints in the lab we still use the *backslash operator* of `MATLAB` to solve the equation system after QR decomposition by steps (f) - (h) of the lab assignments.

The solution will also contain estimated values for the constants c_1 and c_2 but as they are not part of the AR-model used for generation of the signals they should be close to 0.