Linear regression

Bengt Carlsson
Systems and Control
Dept of Information Technology, Uppsala University

August 25, 2014

Abstract

This material is compiled for the course Empirical Modelling. Sections marked with a star ($*$) are not central in the courses. The main source of inspiration when writing this text has been Chapter 4 in the book "System Identification" by Söderström and Stoica (Prentice Hall, 1989) which also may be consulted for a more thorough treatment of the material presented here. The book is available for free download here:
http://www.it.uu.se/research/syscon/Ident

Contents

1 Introduction

Mathematical models are frequently used in both technical and non-technical areas. In this note we will give an overview of one of the most popular model structures, namely linear regression models. In particular we will describe how the parameters in a linear regression model can be fitted to recorded data by the least squares method. Some statistical analyses are also provided

Linear regression has a long history and can be traced back (at least) to Gauss who used such techniques for calculating orbits of planets. Since then, linear regression has been used in numerous applications.

2 The linear regression model

Consider the following model structure:

$$\hat{y}(t) = \varphi_1(t)\theta_1 + \varphi_2(t)\theta_2 + \ldots + \varphi_n(t)\theta_n = \varphi^T(t)\theta \tag{1}$$

where $\hat{y}(t)$ is the output from the model, $\varphi(t) = [\varphi_1(t) \quad \varphi_2(t) \quad \ldots \quad \varphi_n(t)]^T$ is an n-dimensional column vector of *known* variables; "the regressors", and $\theta = [\theta_1 \quad \theta_2 \quad \ldots \quad \theta_n]^T$ is an n-dimensional column vector of *unknown* parameters. The transpose of a vector or matrix is denoted T. The argument $t = 1,\ 2,\ 3\ldots$ is a counter, which very often is used as a time index.

The problem to be discussed is, given a set of measured/observed data, denoted $y(t)$, to find (or rather estimate) the unknown parameter vector θ. The basic idea is to "fit" the model to the data, so that $y - \hat{y}$ becomes small. The model should hence give a good *prediction* of the measured data. Before treating this problem, some examples of linear regression models are given.

2.1 Examples of linear regression models

Below we give a number of examples of models which can be written in the standard form $\hat{y}(t) = \varphi(t)^T\theta$

- The polynomial trend:

  $$\hat{y}(t) = a_o + a_1 t + \ldots + a_n t^n$$

  can be written as $\hat{y}(t) = \varphi(t)^T\theta$ with

  $$\begin{aligned} \varphi(t) &= (1 \quad t \ldots t^n)^T \\ \theta &= (a_o \quad a_1 \ldots a_n)^T \end{aligned}$$

- Sum of exponential functions:

  $$\hat{y}(t) = b_1 e^{-k_1 t} + b_2 e^{-k_2 t} + \ldots + b_n e^{-k_n t}$$

  Assume that $k_1,\ \ k_2 \ldots k_n$ are known. We then have:

  $$\begin{aligned} \varphi(t) &= (e^{-k_1 t} \quad e^{-k_2 t} \ldots e^{-k_n t})^T \\ \theta &= (b_1 \quad b_2 \ldots b_n)^T \end{aligned}$$

- "The gas law". From basic thermodynamics we have

$$\begin{aligned} pV^\gamma &= C \\ p &= V^{-\gamma}C \\ \log p &= -\gamma \log V + \log C \end{aligned}$$

where p is the pressure, V is the volume, γ is the ratio of the specific heat capacities, and C is a constant. Assume that p is measured, V is known, and we want to find γ and C. Let $\hat{y} = \log p$, we may then write the model as a linear regression with

$$\begin{aligned} \varphi(t) &= (-\log V \;\; 1)^T \\ \theta &= (\gamma \;\; \log C)^T \end{aligned}$$

- The FIR-model (Finite Impulse Response)

$$\begin{aligned} \hat{y}(t) &= b_o u(t) + b_1 u(t-1) + \ldots + b_n u(t-n) \\ &\Rightarrow \\ \varphi(t) &= (u(t) \;\; u(t-1) \ldots u(t-n))^T \\ \theta &= (b_o \;\; b_1 \ldots b_n)^T \end{aligned}$$

where u is the input signal and $b_o, \;\; \ldots \;\; b_n$ are the unknown parameters. The basic FIR models can easily be expanded to cover more than one input signal. One example of a FIR model with two input signals (u_1 and u_2) is:

$$\begin{aligned} \hat{y}(t) &= b_{1,o} u_1(t) + b_{1,1} u_1(t-1) + \ldots + b_{1,n} u_1(t-n) \\ &\quad + b_{2,o} u_2(t) + b_{2,1} u_2(t-1) + \ldots + b_{2,n} u_2(t-n) \end{aligned}$$

Make sure that you can write this model in the linear regression form (1).

- The ARX-model (AutoRegressive model with an eXternal input)

$$\begin{aligned} \hat{y}(t) &= -a_1 y(t-1) - a_2 y(t-2) - \ldots - a_{na} y(t-na) \\ &\quad + b_o u(t) + b_1 u(t-1) + \ldots + b_{nb} u(t-nb) \\ &\Rightarrow \\ \varphi(t) &= (-y(t-1) \;\; -y(t-2) \ldots -y(t-na) \;\; u(t) \;\; u(t-1) \ldots u(t-nb))^T \\ \theta &= (a_1 \;\; a_2 \ldots a_{na} \;\; b_o \;\; b_1 \ldots b_{nb})^T \end{aligned}$$

The output from the model is based on old measured data ($y(t-1)$ etc) and old input signals. This model is one of the most used models for estimating dynamical systems. It will be described in more detail later in the course.

3 The least squares method (Minsta kvadratmetoden)

Assume that a data set $\{y(t), \varphi(t)\}_{t=1,...,N}$ from a system has been collected. That is, we have N samples from the system. Assume also that the system can be described by a linear regression model (of known structure but unknown parameters). The problem then is to find a "good" estimate of the unknown vector θ given the measurements. Basically, we want to "fit the model to the data" as good as possible.

3.1 The loss function

In the *least squares method*, the following loss function is to be minimised with respect to θ:

$$V(\theta) = \sum_{t=1}^{N} (y(t) - \hat{y}(t))^2 = \sum_{t=1}^{N} (y(t) - \varphi^T(t)\theta)^2 \tag{2}$$

That is, we seek for a model that could "predict" the real data as good (in a mean squared sense) as possible. Note that the classical approach "fitting a linear trend" is one example of the least squares method.

It is natural to introduce the equation (or prediction) errors as $\epsilon(t) = y(t) - \hat{y}(t)$. The loss function (2) can then be written as

$$V(\theta) = \sum_{t=1}^{N} \epsilon(t)^2$$

Remarks:

- If the data $y(t)$ would be noise-free a natural choice would be to let $N = n$, where n is the number of unknown parameters (that is the size of θ). In practice, this is seldom the case! For example, to only use two points to fit a linear trend would in most cases be very "dangerous". It hence seems reasonable to chose $N \gg n$. The choice of N will determine the accuracy: a higher N gives more accurate model parameters (if the chosen model structure is correct). The choice of N is to be discussed more in the next section.

- It is common to write the loss function in normalised form as

$$V(\theta) = \frac{1}{N} \sum_{t=1}^{N} (y(t) - \hat{y}(t))^2$$

  Obviously, this gives the same minimizing θ.

3.2 The least squares estimate

In this section we will give the solution to the least squares problem, that is, to give an expression for the vector θ that minimizes the least squares criterion (2).

We present the result as a Theorem:

Theorem 1. Assume that the matrix $R_N = \sum_{t=1}^{N} \varphi(t)\varphi^T(t)$ is invertible. The θ that minimises $V(\theta)$ in (2) is given by :

$$\hat{\theta} = \left[\sum_{t=1}^{N} \varphi(t)\varphi^T(t)\right]^{-1} \sum_{t=1}^{N} \varphi(t)y(t) \tag{3}$$

Proof. The proof is to simply set the first derivative (with respect to θ) of the loss function to zero (or more precisely to a zero vector) and solve for the parameter vector.

$$\begin{aligned} \frac{\delta V(\theta)}{\delta \theta} &= 2\sum_{t=1}^{N}(y(t) - \varphi^T(t)\theta)\frac{-\delta\varphi^T(t)\theta}{\delta\theta} = -2\sum_{t=1}^{N}(y(t) - \varphi^T(t)\theta)\varphi(t) \\ &= -2\sum_{t=1}^{N}(\varphi(t)y(t) - \varphi(t)\varphi^T(t)\theta) \end{aligned}$$

In the last equality we have put $\varphi(t)$ in front of the scalars $y(t)$ and $\varphi^T(t)\theta$. Next we set the derivative equal to the zero vector:

$$\frac{\delta V(\theta)}{\delta \theta} = 0$$

which gives

$$\sum_{t=1}^{N} \varphi(t)y(t)) = \sum_{t=1}^{N} \varphi(t)\varphi^T(t)\hat{\theta}$$

Finally, solving for $\hat{\theta}$ (by multiplying with the inverse of R_N from the left) gives the least squares estimate (3). □

Remarks:

- For *all* models that can be written as a linear regression, we have an analytical solution to the least squares problem (a few examples are given later). The expression (3) is very easy to calculate with modern software (for example using Matlab) where also good numerical methods for calculating this estimate are implemented. See also Section 3.3.

- To calculate ordinary least squares estimate (3) it is crucial that R_N is invertible. Situations when this does not hold (or almost does not hold) are referred to *ill-conditioned* linear regression problems. This situation appears when the regressors are colinear (or nearly colinear) which means that two rows or columns of the matrix R_N is (almost) similar. This is a common problem in many practical applications! One way to solve ill conditioned linear regression problems is to "remove" the part in R_N that causes the colinearity. This can, for example, be done by singular value decomposition. We will not treat this problem here, but it will be discussed later in the course.

- It is common to write the least squares estimate in normalised form

$$\hat{\theta} = \left[\frac{1}{N}\sum_{t=1}^{N}\varphi(t)\varphi^T(t)\right]^{-1}\frac{1}{N}\sum_{t=1}^{N}\varphi(t)y(t)$$

  This gives of course the same solution as (3) but often simplifies the analysis when considering the case when the number of data points goes to infinity.

3.3 A matrix formulation

In this Section we will give a matrix presentation of the least squares method. It is assumed (as before) that N measurements of $y(t)$ and $\varphi(t)$ are available. These measurements can be written as a vector and a matrix by employing the notations:

$$Y = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}$$

$$\Phi = \begin{bmatrix} \varphi^T(1) \\ \vdots \\ \varphi^T(N) \end{bmatrix}$$

Similarly, the predicted output from the linear regression model

$$\hat{y}(t) = \varphi(t)^T\theta, \quad t = 1, \ldots, N$$

can be written

$$\hat{Y} = \Phi\theta$$

where

$$\hat{Y} = \begin{bmatrix} \hat{y}(1) \\ \vdots \\ \hat{y}(N) \end{bmatrix}$$

The loss function (2) can now be written

$$V(\theta) = (Y - \Phi\theta)^T(Y - \Phi\theta)$$

The θ that minimises $V(\theta)$ is given by :

$$\hat{\theta} = \left[\sum_{t=1}^{N}\varphi(t)\varphi^T(t)\right]^{-1}\sum_{t=1}^{N}\varphi(t)y(t)) = [\Phi^T\Phi]^{-1}\Phi^TY \tag{4}$$

The last equality follows directly from

$$\sum_{t=1}^{N}\varphi(t)\varphi^T(t) = [\varphi(1)...\varphi(N)]\begin{bmatrix} \varphi^T(1) \\ \vdots \\ \varphi^T(N) \end{bmatrix} = \Phi^T\Phi$$

and

$$\sum_{t=1}^{N}\varphi(t)y(t)) = [\varphi(1)...\varphi(N)]\begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} = \Phi^TY$$

Remarks:

- The least square estimate (4) is very easy to calculate in Matlab. You do not need to write the whole expression in (4). Assume that the data matrix Φ is stored in the variable *Phi* in Matlab. The least squares estimate is then (note the use of the backslash operator) given by

```
>> theta_hat=Phi\Y
```

- In practice we may not have data to fill up the first rows in the Φ vector. The trick is then to cut the rows that can not be filled with known data (and also to cut the rows in the Y vector accordingly). This approach corresponds to using the following loss function

$$V(\theta) = \sum_{t=n1}^{N} (y(t) - \varphi^T(t)\theta)^2 \tag{5}$$

where $n1$ is chosen so $\varphi(n1)$ contains known data (for example $u(1)$ $u(2)$). Note that using $n1 = 1$ for a FIR model would require $u(0)$ $u(-1)$ etc which are NOT known. The choice of $n1$ depends on the model order.

3.4 Proof of the least squares estimate using matrix notations*

In this Section we will give an alternative proof of the least squares estimate using the matrix formulation. The loss function may be rewritten as

$$V(\theta) = (Y - \Phi\theta)^T(Y - \Phi\theta) = Y^TY - Y^T\Phi\theta - \theta^T\Phi^TY + \theta^T\Phi^T\Phi\theta$$

By applying rules for differentiation of matrices (see Appendix A) the gradient can be written as

$$\frac{\delta V(\theta)}{\delta\theta} = -Y^T\Phi - Y^T\Phi + 2\theta^T\Phi^T\Phi = 2(\theta^T\Phi^T\Phi - Y^T\Phi)$$

By setting the transpose of the gradient to zero

$$\frac{dV(\theta)}{d\theta}|^T = 0$$

it is seen that the least squares estimate is given by

$$\hat{\theta} = [\Phi^T\Phi]^{-1}\Phi^TY$$

3.5 A simple example

The simplest example of a linear regression model is

$$\hat{y}(t) = m$$

This means that a constant m should be estimated from a number of measurements. The model is represented by the standard notation $\varphi(t) = 1$ and $\theta = m$ (see (1)). Assume that N data points, $y(1), y(2)$ ... $y(N)$ are available. We want to estimate the constant m from available data using the least squares method. The estimate is given by

$$\hat{\theta} = \left[\sum_{t=1}^{N} \varphi(t)\varphi^T(t)\right]^{-1} \sum_{t=1}^{N} \varphi(t)y(t) = \left[\sum_{t=1}^{N} 1\right]^{-1} \sum_{t=1}^{N} 1 \cdot y(t) = \frac{1}{N}\sum_{t=1}^{N} y(t)$$

which we recognize as the arithmetic mean of the measurements.

3.6 Nonlinear regression models

The least squares method can also be used for nonlinear models. Consider the model structure

$$\hat{y}(t) = g(\varphi(t), \theta)$$

where g is some nonlinear functions. In order to estimate θ, the following loss function can be used

$$V(\theta) = \sum_{t=1}^{N} [y(t) - g(\varphi(t), \theta)]^2$$

We then search for θ that minimises $V(\theta)$. This can be formally written as

$$\hat{\theta} = \arg\min_{\theta} V(\theta)$$

Remarks:

- In general no analytical solution exists instead a numerical (iterative) search must be used.
- When using a numerical search routine, there is, in general, no guarantee that the global minima is found. The search routine may stop at a local minima.
- In Matlab the function *fminsearch* can be used for solving nonlinear regression problems. There is also a toolbox, Curve Fitting Toolbox.

4 Analysis

Above we have shown how the parameters in a linear regression model can be estimated by the least squares method. An important question is what accuracy the estimated parameters have. In order to answer such a question we need to make assumptions on how the data ($y(t)$) are generated. In general, the accuracy of the estimate will depend on the "noise corruption" in the data as well as on the number of data points used in the estimation.

4.1 The white noise case

In this Section we will treat the case when the noise is white. The accuracy result is based on a number of assumptions that is presented next:

Assumption A1.
Assume that the data are generated by ("the true system"):

$$y(t) = \varphi^T(t)\theta_o + e(t) \quad t = 1, \ldots, N \tag{6}$$

where $e(t)$ is a nonmeasurable disturbances term to be specified below. In matrix form, (6) reads

$$Y = \Phi\theta_o + \mathbf{e} \tag{7}$$

where $\mathbf{e} = [e(1) \ \ldots e(N)]^T$.

Assumption A2.
It is assumed that $e(t)$ is a white noise process[1] with variance λ.

Assumption A3.
It is finally assumed that $E\{\varphi(t)e(s)\} = 0$ for all t and s. This means that the regression vector is not influenced (directly or indirectly) by the noise source $e(t)$

Assumption A3 simplifies the analyses considerable. When taking expectation with respect to $e(t)$ we have, for example,

$$E\{\Phi^T\Phi\mathbf{e}\} = \Phi^T\Phi E\{\mathbf{e}\}.$$

Theorem 2. If Assumptions A1-A3 hold then

1. The least squares estimate $\hat{\theta}$ is an unbiased estimate of θ_o, that is $E\{\hat{\theta}\} = \theta_o$.

2. The variance of the parameters $\hat{\theta}_i$, $i = 1, 2, \ldots n$ in the estimated parameter vector $\hat{\theta}$ is given by

   $$P = \text{cov}\ \hat{\theta} = E\{(\hat{\theta} - E\hat{\theta})(\hat{\theta} - E\hat{\theta})^T\} = E\{(\hat{\theta} - \theta_o)(\hat{\theta} - \theta_o)^T\} = \lambda(\Phi^T\Phi)^{-1} \quad (8)$$

   In particular we have for the i th parameter, $\text{var}\hat{\theta}(i) = P(i,i)$, that is the variance of the parameters can be found by inspecting the diagonal elements of the covariance matrix P The uncertainty of the least squares estimate as expressed by the covariance matrix P is given by

   $$P = \text{cov}\ \hat{\theta} = E\{(\hat{\theta} - E\hat{\theta})(\hat{\theta} - E\hat{\theta})^T\} = E\{(\hat{\theta} - \theta_o)(\hat{\theta} - \theta_o)^T\} = \lambda(\Phi^T\Phi)^{-1} \quad (9)$$

3. An unbiased estimate of the variance of the noise λ is given by

   $$\hat{\lambda} = \frac{1}{N-n} V(\hat{\theta}) \quad (10)$$

   where n is the number of parameters ($n = \dim\theta$) and N is the number of data points.

Proof.

1. Proof of unbiasness:

   $$E\{\hat{\theta}\} = E\{[\Phi^T\Phi]^{-1}\Phi^T Y\} = E\{[\Phi^T\Phi]^{-1}\Phi^T(\Phi\theta_o + \mathbf{e})\} = \theta_o + [\Phi^T\Phi]^{-1}\Phi^T\Phi E\{\mathbf{e}\} = \theta_o$$

   The last equality follows from the assumption (A2) that the mean value of $e = 0$. Note that it is only required that $E\{e(t)\} = 0$ for the estimate to be unbiased.

[1] A white noise process $e(t)$ is a sequence of random variables that are *uncorrelated*, have mean zero, and a constant finite variance. Hence, $e(t)$ is a white noise process if $E\{e(t)\} = 0$, $E\{e^2(t)\} = \lambda$, and $E\{e(t)e(j)\} = 0$ for t not equal to j.

2. Proof of the covariance expression:
   First, note that

$$\begin{aligned}\hat{\theta} &= [\Phi^T\Phi]^{-1}\Phi^T Y = [\Phi^T\Phi]^{-1}\Phi^T(\Phi\theta_o + \mathbf{e}) = [\Phi^T\Phi]^{-1}\Phi^T\Phi\theta_o + [\Phi^T\Phi]^{-1}\Phi^T\mathbf{e} \\ &= \theta_o + [\Phi^T\Phi]^{-1}\Phi^T\mathbf{e}\end{aligned}$$

   Hence, $\hat{\theta} - \theta_o = [\Phi^T\Phi]^{-1}\Phi^T\mathbf{e}$. We also have that $E\{\hat{\theta}\} = \theta_o$. We then have

$$\begin{aligned}\text{cov}\hat{\theta} &= E\{(\hat{\theta} - \theta_o)(\hat{\theta} - \theta_o)^T\} = E\{[\Phi^T\Phi]^{-1}\Phi^T\mathbf{e}([\Phi^T\Phi]^{-1}\Phi^T\mathbf{e})^T\} \\ &= E\{[\Phi^T\Phi]^{-1}\Phi^T\mathbf{e}\mathbf{e}^\mathbf{T}\Phi[\Phi^T\Phi]^{-1}\}\end{aligned}$$

   Now, using Assumption A3 the only expectation that is needed to calculate is $E\{\mathbf{e}\mathbf{e}^\mathbf{T}\} = E\{[e(1);\ \ldots e(N)]^T[e(1);\ \ldots e(N)]\}$. Further, since the noise is assumed to be white, this matrix will be a diagonal matrix, where every diagonal element has the value λ. Hence, $E\{\mathbf{e}\mathbf{e}^\mathbf{T}\} = \lambda I$ where I is the identity matrix. We thus get

$$\text{cov}\hat{\theta} = [\Phi^T\Phi]^{-1}\Phi^T\lambda I\Phi[\Phi^T\Phi]^{-1} = \lambda[\Phi^T\Phi]^{-1}\Phi^T\Phi[\Phi^T\Phi]^{-1} = \lambda[\Phi^T\Phi]^{-1}$$

3. The proof that $\hat{\lambda} = \frac{1}{N-n}V(\hat{\theta})$ is omitted here, but can be found in the book System Identification (see the Abstract).

□

An important use of theorem 2
Let $\hat{\theta}_i$, $i = 1,\ \ldots n$ denotes the i'th component in the vector $\hat{\theta}$. We then have

$$\text{var}(\hat{\theta}_i) = P_{i,i} \quad i = 1,\ \ldots n \tag{11}$$

where $P_{i,i}$ denotes the i'th diagonal element in P. Hence the parameters can be given a "quality tag". The covariance matrix P can be estimated from data using

$$\hat{P} = \hat{\lambda}(\Phi^T\Phi)^{-1}$$

where $\hat{\lambda}$ is obtained from (10).

Remarks:

- Note that we have assumed in Assumption A1 that the true system and the model have the same structure ($\varphi(t)$ is the same). For example, if the model is a linear trend we assume that the true system that has generated the data is also a linear trend (but with an additional noise term).
- The signal $e(t)$ in (6) typically represents measurement noise and/or process disturbances and is commonly called "noise".
- When A3 holds, it is common to say that $\varphi(t)$ is deterministic.
- If we assume that the noise e has a Gaussian distribution, $\hat{\theta}$ will also be Gaussian

$$\hat{\theta} \in \mathcal{N}(\theta_o, P)$$

and

$$\frac{\hat{\theta}(i) - \theta_o(i)}{\sqrt{P(i,i)}} \in \mathcal{N}(0,1)$$

Hence the probability that $\hat{\theta}(i)$ deviates from $\theta_o(i)$ with more than $\alpha\sqrt{P(i,i)}$ is the $(1-\alpha)$-level of the normal distribution which is available in standard statistical tables.

- One very important example when Assumption 3 does **not** hold is for the ARX model (see Section 2.1). The ARX case will be analyzed later in the course.

4.2 Accuracy of first order FIR model

Consider the following model

$$\hat{y}(t) = bu(t)$$

This corresponds to $\varphi(t) = u(t)$ and $\theta = b$. Assume that N data pairs, $y(1), u(1),\ \ldots\ y(N), u(N)$ are available. The least squares estimate is then given by

$$\hat{\theta} = \left[\sum_{t=1}^{N} \varphi(t)\varphi^T(t)\right]^{-1} \sum_{t=1}^{N} \varphi(t)y(t) = \left[\sum_{t=1}^{N} u^2(t)\right]^{-1} \sum_{t=1}^{N} u(t)y(t) = \frac{1}{\sum_{t=1}^{N} u^2(t)} \sum_{t=1}^{N} u(t)y(t)$$

If the assumptions used in Theorem 2 are fulfilled the variance of $\hat{b} = \hat{\theta}$ is given by:

$$\mathrm{var}(\hat{b}) = \frac{\lambda}{\sum_{t=1}^{N} u^2(t)}$$

The variance decreases if

1. The number of data N increases,
   or
2. The input signal energy is increased,
   or
3. The noise level λ is decreased.

In general these three factors (number of data points, signal energy, and noise level) will affect the quality of the estimated parameters.

4.3 Non white noise

Here, we will consider the case when the noise in not white. Assume that Assumptions A1 and A3 hold but Assumption A2 is generalized to:

Assumption A2gen.
The noise ("measurement noise") $e(t)$ in (6) has zero mean value but may be correlated so that $E\{e(t)e(s)\} \neq 0$. Then we write $E\{ee^T\} = R$ where R is a symmetric matrix describing the correlation of the noise.

Theorem 3. Assume that A1, A2gen and A3 hold, then

1. The least squares estimate $\hat{\theta}$ is (still) an unbiased estimate of θ_o, that is $E\{\hat{\theta}\} = \theta_o$.

2. The covariance matrix of the least squares estimate is

$$\text{cov}\hat{\theta} = (\Phi^T\Phi)^{-1}\Phi^T R\Phi(\Phi^T\Phi)^{-1} \tag{12}$$

3. If the noise correlation matrix R is known, a more accurate estimate than the standard least squares estimate is possible. The method is known as "BLUE" (Best Linear Unbiased Estimate) and is given by:

$$\hat{\theta}_{BLUE} = (\Phi^T R^{-1}\Phi)^{-1}\Phi^T R^{-1}Y \tag{13}$$

The covariance matrix of the BLUE is

$$\text{cov}\hat{\theta}_{BLUE} = (\Phi^T R^{-1}\Phi)^{-1} \tag{14}$$

Remarks:

- It is worth stressing that the results in Theorem 3 do **not** hold for the ARX model (see Section 2.1). In fact, for an ARX model, the estimated parameters will be biased $E\{\hat{\theta}\} \neq \theta_o$ even when the number of data points goes to infinity.
- The BLUE estimate (13) can be calculated with the Matlab function *lscov*.

Proof.

1. Proof of unbiasness: The result follows directly from the proof of Theorem 2 since the mean value of the noise is still zero.

2. Proof of the covariance expression:
   Following the proof of Theorem 2 we have

$$\begin{aligned} \text{cov}\hat{\theta} &= E\{(\hat{\theta} - \theta_o)(\hat{\theta} - \theta_o)^T\} = E\{[\Phi^T\Phi]^{-1}\Phi^T\mathbf{e}([\Phi^T\Phi]^{-1}\Phi^T\mathbf{e})^T\} \\ &= E\{[\Phi^T\Phi]^{-1}\Phi^T\mathbf{e}\mathbf{e}^\mathbf{T}\Phi[\Phi^T\Phi]^{-1}\} \end{aligned}$$

   Now using $E\{\mathbf{e}\mathbf{e}^\mathbf{T}\} = R$ the results follows directly

3. The proof that BLUE has a lower covariance matrix than the least squares method is omitted here, but can be found in the book System Identification (see the Abstract).

□

4.4 Example of the best linear unbiased estimate (BLUE)

Consider again the simple model

$$\hat{y}(t) = m$$

Assume that the data are obtained from the system

$$\hat{y}(t) = m_o + e(t)$$

where noise is independent, has zero mean $Ee(t) = 0$, but a varying variance $Ee^2(t) = \lambda(t)$. It is assumed that $\lambda(t)$ is known. Assuming N measurements of y, then

$$\begin{aligned} \Phi &= \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \\ R &= \begin{bmatrix} \lambda(1) & 0 & \dots & 0 \\ 0 & \lambda(2) & \dots & 0 \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \lambda(N) \end{bmatrix} \end{aligned}$$

The BLUE (see 13) of $\theta_o = m_o$ is given by

$$\hat{\theta} = (\Phi^T R^{-1} \Phi)^{-1} \Phi^T R^{-1} Y = \frac{1}{\sum_{j=1}^{N} \frac{1}{\lambda(j)}} \sum_{i=1}^{N} \frac{1}{\lambda(i)} y(i)$$

This is a weighted arithmetic mean of the measurements. The weight of $y(i)$ in the estimate is given by

$$\hat{\theta} = (\Phi^T R^{-1} \Phi)^{-1} \Phi^T R^{-1} Y = \frac{1}{\sum_{j=1}^{N} \frac{1}{\lambda(j)}} \frac{1}{\lambda(i)}$$

This weight is small if a measurement is inaccurate (meaning that $\lambda(i)$ is large) and vice versa. Note also that if the noise variance is constant $\lambda(t) = \lambda$ then the BLUE estimate becomes the ordinarily least squares estimate in the example shown in Section 3.5

5 On the choice of model structure and model order

An very crucial question in system identification is how to choose the model structure and the model order. Later in the course, this will be treated in detail. Let us here, just give a few remarks. We will also briefly describes the most common method to check the model quality, namely the the coefficient of determination R^2.

Concerning the choice of model structure, two options are available:

1. Use physical insights. In some cases, knowledge of the system may give hints of a suitable model structure.

2. Try different model structures and use the one which could describe the data "best" (or sufficiently good for the intended use of the model).

We need also to determine the model order (n=dim θ). In practise, this is done by a combination of statistical test and common sense. Note that it is not a good idea to try to find a model order that minimises the loss function (2) since the loss function will decrease as the model order is increased. This will be illustrated in the first computer laboratory work.

5.1 The coefficient of determination

The coefficient of determination R^2, gives information of about the goodness of fit of a model and is calculated as

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{15}$$

where

$$SS_{res} = \sum_{t=1}^{N} (y(t) - \hat{y}(t))^2 \tag{16}$$

$$SS_{tot} = \sum_{t=1}^{N} (y(t) - \bar{y})^2 \tag{17}$$

with $\bar{y} = \frac{1}{N} \sum_{t=1}^{N} y(t)$.

The coefficient of determination is the percent of the variation that can be explained by the model and is the ratio between the explained variation and the total variation. The closer the value of R^2 is to 1, the better the regression is. An R^2 of 1.0 indicates that the regression line perfectly fits the data. But remember "Correlation does not imply causation".

Remarks:

- In the System Identification Toolbox, R^2 is called `FIT` and is given in percentage.

6 Principal component analysis*

Finally, we will give a very brief introduction to Principal Component Analysis (PCA). This is a method which is used very often in practice (including several recent Master thesis) and will also be a key topic in the first guest lecture.

PCA is a method to extract relevant information from a data set without using a (explicit) model. In the following we will assume that the data has been organized in a $m|n$ matrix X. Typically, m is the number of measurements and n is the number of dependant variables (for example sensors). The data must not come from a technical process, it may for instance consists of demographic data where m is the number of studied countries and n is the number of indicator variables.

Each row may then consists of a countries BNP, population etc.

Frequently, one wants to find possible pattern in the data, and highlight differences and similarities. This is not a trivial problem if m and n is large. One typical example is process monitoring, suppose that we have measured a number of process variables (pH, temperature etc) and want to find which variables are related to each other. Or under what conditions are the process running in an optimal way. This may be hard to find out by only looking at one variable (a column in the X matrix) at the time. It may very well be so that an inspection of each variable separately does not reveal any useful information.

PCA is a way to reduce the dimensionality of the problem (the X matrix). It is obvious that with increasing dimensionality the data will be harder and harder to visualize and to interpret. But "the underlying" dimension may be much smaller than n. Consider for example, $n = 3$, this will correspond to a "cloud" in the 3D space, where each point represent a measurement. It might be so that all points residue close to a common plane, that is the underlying dimension is two and not three. What PCA basically does is to make a coordinate transformation of the X matrix where the first coordinate axes gives the direction where the data contain most information (variance), the second coordinate gives the second most important direction etc. Hence we can see how many directions (dimension) that is needed in order to describe the data sufficiently well. Not seldom, only two dimension is needed, and it is then possible to plot the transformed data in a 2D plot.

Mathematically, PCA is a method of writing a matrix X a sum of r matrices of rank 1:

$$X = M_1 + M_2 + ; \ \ldots \ + M_a + E$$

where E is the residual error matrix. The error $E = 0$ if $a = r$, where r is the rank of the matrix X. In practice, a is chosen so that E is sufficiently small. The matrices M_i can be written as the product of two vectors t_i and p_i as follows

$$X = t_1 p_1^T + t_2 p_2^T ; \ \ldots \ + t_a p_a^T + E$$

or

$$X = TP^T + E$$

where T is made up from the t_i's as columns and P is made up from the p_i's as columns. The vectors t_i are called scores and the vectors p_i are called loadings. To find T and P we can use a singular value decomposition (SVD) on X:

$$X = UDV$$

It can be shown that $T = UD$ and $P = V$.

SVD is easy to do in for example Matlab. Dedicated toolboxes exists for more advanced studies, for example the "PLS Toolbox" by Eigenvector.

One important aspect of PCA is that it is not scaling independent. If a variable is measured in grams or kilo hence affect the results. In order to cope with this

the data must be scaled before the PCA. The standard procedure is to remove the mean values from each column (mean centering) and divide each colum by the column variance.

The literature on PCA (and related methods) is huge! By a web search you may found numerous applications as well as detailed descriptions.

7 Concluding remarks

In this note we have given an introduction to linear regression and in particular how the parameters in a linear regression model can be estimated with the least squares method. The ideas presented here form the basics for "System identification methods". We have seen that many types of models can be written as linear regressions. The statistical analyses show how estimated models can be given a "quality tag" (model accuracy in terms of variance). It has been noted though that the very much used ARX-model does not fulfill the Assumptions used in the derived accuracy results and hence requires a separate investigation (to be covered later in the course).

A Some matrix algebra

Here some important results from matrix algebra are summarised.

- For a symmetric matrix P it holds that $P = P^T$
- $(AB)^T = B^T A^T$
- Some properties of a positive definite matrix P (commonly written as $P > 0$):
  - $x^T P x > 0$ for all $x > 0$.
  - All eigenvalues of P is larger than zero.
  - $\det P > 0$

Differentiation, let u be a column vector, then

$$\begin{aligned}
\frac{d}{du} u^T P u &= 2u^T P \quad \text{if } P \text{ symmetric} \\
\frac{d}{du} z^T B u &= z^T B \quad \text{z = vector, B = matrix} \\
\frac{d}{du} u^T B z &= z^T B^T
\end{aligned}$$

B Några grundläggande statistiska begrepp

- Fördelningsfunktion för den stokastiska variabeln (s.v.) X defineras enligt $F_X(x) = P(X \leq x)$, dvs sannolikheten att den s.v. X är mindre än eller lika med talet x.
- Täthetsfunktion $f_X(x)$ = derivatan av fördelningsfunktionen. Vi har att

$$P(a < X \leq b) = \int_a^b f_X(t)dt$$

- Normalfördelning $X \in N(m, \lambda)$ har täthetsfunktionen

$$f_X(x) = \frac{1}{\sqrt{2\pi\lambda}} e^{-(x-m)^2/(2\lambda)}$$

  där m =medelvärdet och λ =variansen.
- Om $f_{X,Y}(x, y) = f_X(x)f_Y(y)$ är de s.v. X och Y oberoende.
- Väntevärde (medelvärde): $m = E\{X\} = \int_{-\infty}^{\infty} x f_X(x)dx$. E är en linjär operator: $E\{aX + b\} = aE\{X\} + b$.
- Varians[2]: $V(X) = E\{(X - m)^2\}$. Vi har

$$\begin{aligned} V(X) &= E\{X^2\} - (E\{X\})^2 \\ V(aX + b) &= a^2 V(X) \end{aligned}$$

- Kovarians: $\mathrm{Cov}(X, Y) = E\{(X - m_X)(Y - m_Y)\}$.
- Om $\mathrm{Cov}(X, Y) = 0$ är X och Y okorrelerade. Notera att obereoende medför okorrelerade (men inte tvärtom).

[2]Standardavvikelse är kvadratroten av variansen.

Något om parameterskattningar (punktskattning)

- Om $X_1, \ldots X_N$ är oberoende och normalfördelade s.v. $N(m, \lambda)$ så är följande skattning av medelvärdet

$$\hat{m} = \frac{1}{N} \sum_{i=1}^{N} X_i$$

  normalfördelad och väntevärdesriktig dvs $E\{\hat{m}\} = m$. Skattningens varians ges av $V(\hat{m}) = \frac{\lambda}{N}$.

- Låt $\hat{\theta}(N)$ vara en skattning av den okända parametrn θ_o givet N stycken observationer.Följande definitioner är centrala:
  - Skattningens bias ges av $b = E\{\hat{\theta}(N)\} - \theta_o$
  - Skattningens varians $v = E\{(\hat{\theta}(N) - E\{\hat{\theta}(N)\})^2\}$
  - Medelkvadratfelet[3] MSE= $v + b^2$
  - Asymptotisk väntevärdesriktig: $E\{\hat{\theta}(N)\} \to \theta_o$ då $N \to \infty$.
  - Konsistens: $E\{(\hat{\theta}(N) - \theta_o)^2\} \to 0$ då $N \to \infty$.

- Flerdimensionella s.v.

  Låt $X = [X_1, X_2, \ldots, X_n]^T$ vara en n-dimensionell s.v. Då ges väntevärdet av $m = E\{X\} = [EX_1, EX_2, \ldots, EX_n]^T$. Kovarianvariansmatrisen definieras $V(X) = E\{(X-m)(X-m)^T\}$ vilket är en symmetrisk och positiv semidefinit $n|n$ matris. Om $Y = c + AX$ är $E\{Y\} = c + AE\{X\}$ och $V_Y = AV(X)A^T$. En linjärkombination av en n-dimensionell normalfördelad s.v är också normalfördelad.

[3]Engelska: MSE-Mean Squared Error