System Identification, Lecture 2

Kristiaan Pelckmans (IT/UU, 2338)

Course code: 1RT875, Report code: 61806,
F, FRI Uppsala University, Information Technology

20 January 2010

Lecture 2

- Linear Regression
- The Least Squares Method
- Properties of the (deterministic) Least Squares Estimator
- BLUE
- Computational Aspects

Linear Regression

SI Procedure: (a) collect data, (b) choose a model class, (c) find the best model in the model class, (d) model validation.

- Linear regression models
  - Conceptionally simple
  - Simple to analyze
  - Simple to implement
  - Low memory consumption
  - Common in Signal Processing. Ex. Echo cancellation
- Original work by Gauss 1809
- Starting point of SI

Linear Regression (Ct'd)

- Model Structure ($\mathcal{M}$):

$$y_\theta(t) = \sum_{i=1}^{n} \theta_i \varphi_i(t) = \varphi(t)^T \theta$$

  where $y_\theta(t) \in \mathbb{R}$ is the model output at time t; $\varphi(t) \in \mathbb{R}^n$ is a (column) vector of known quantities; and $\theta \in \mathbb{R}^n$ is a (column) vector of unknown 'parameters'.

- Stacking up the model for $t = 1, \ldots, N$ gives

$$\mathbf{Y}_\theta = \Phi\theta$$

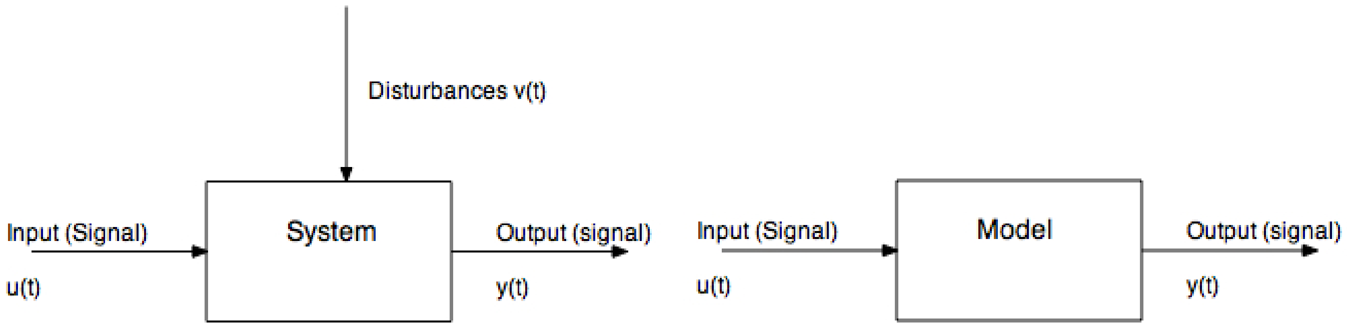
  with $\mathbf{Y}_\theta = \begin{bmatrix} y_\theta(1) \\ \vdots \\ y_\theta(N) \end{bmatrix} \in \mathbb{R}^N$ and $\Phi = \begin{bmatrix} \varphi(1)^T \\ \vdots \\ \varphi(N)^T \end{bmatrix} \in \mathbb{R}^{N \times n}$.

- Linear regression can also be used for certain nonlinear models

Linear Regression (Ct'd)

Problem Find an estimate of θ given measurements $\varphi(1), y(1), \ldots, \varphi(t), y(t), \ldots$ satisfying

$$y(t) = y_\theta(t) + v(t) = \varphi(t)^T\theta + v(t)$$



- Noiseless case ($v(t) = 0$, $\mathcal{M} = \mathcal{S}$). exact Solution exists.
- What to do when noise $v(t) \neq 0$ and/or $\mathcal{M} \neq \mathcal{S}$.

Least Squares (Optimization)

Introduce the equation error

$$\epsilon_\theta(t) = y(t) - y_\theta(t) = y(t) - \varphi(t)\theta,\ t = 1, \ldots, N$$

or compactly

$$\epsilon_\theta = \mathbf{Y} - \mathbf{Y}_\theta$$

Least Squares Method: Choose θ such that $\|\epsilon_\theta\|_2$ is smallest

$$\hat{\theta}_{LS} = \operatorname*{argmin}_\theta V(\theta), \quad V(\theta) = \frac{1}{2}\sum_{t=1}^{N} \epsilon_\theta^2(t) = \frac{1}{2}\epsilon_\theta^T \epsilon_\theta$$

Results: Assume that $\Phi^T\Phi$ is invertible, then

$$\hat{\theta}_{LS} = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{Y}.$$

This estimator is known as the Ordinary Least Squares (OLS) estimator.

Weighted Least Squares estimate: Let $\mathbf{W} = \mathbf{W}^T \in \mathbb{R}^{n\times n}$ be a positive definite symmetric matrix, then

$$\hat{\theta}_{WLS} = \operatorname*{argmin}_{\theta} V_{\mathbf{W}}(\theta), \quad V_{\mathbf{W}}(\theta) = \frac{1}{2}\epsilon_\theta^T\mathbf{W}\epsilon_\theta$$

when $(\Phi^T\mathbf{W}\Phi)$ is invertible, the solution becomes

$$\hat{\theta}_{WLS} = (\Phi^T\mathbf{W}\Phi)^{-1}\Phi^T\mathbf{W}\mathbf{Y}.$$

Note: when $\mathbf{W} = \mathcal{I}_n$, $\theta_{LS} = \theta_{WLS}$.

Least Squares (Geometric Approach)

- Model:

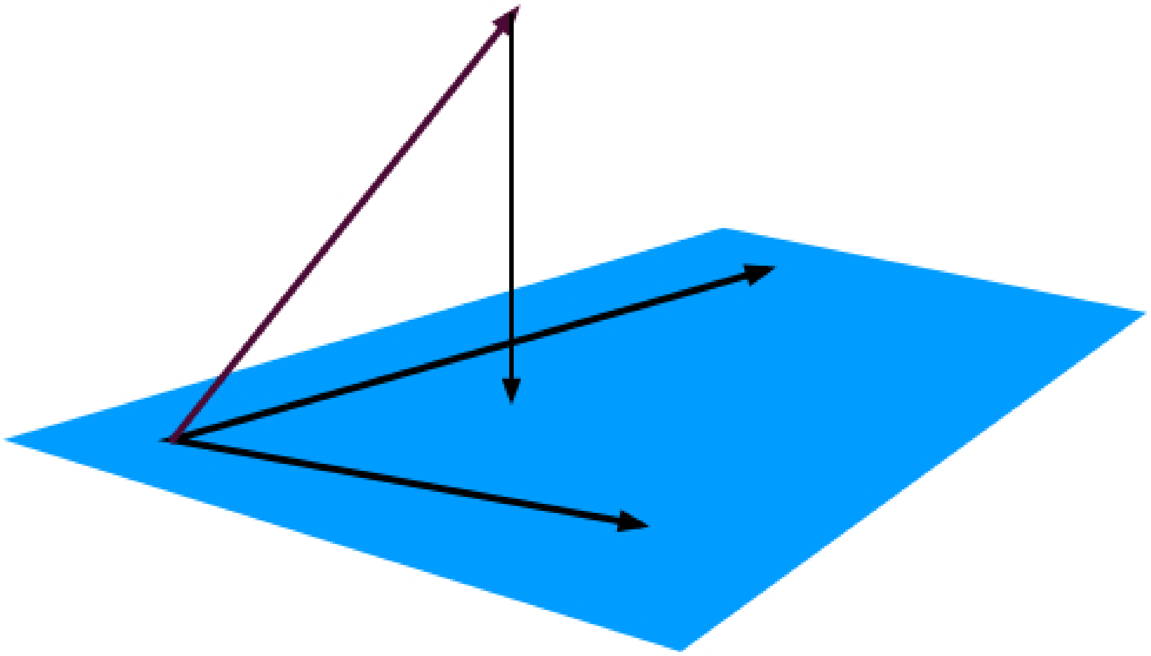
$$Y_\theta = \Phi\theta = \left[\Phi_1, \ldots \Phi_n\right]\theta$$

- Measurements $\mathbf{Y}$.

- $\mathbf{Y}$ and Φ_i are vectors in the vector space $\mathbb{R}^N$

- **Objective:** Find a linear combination of the vectors Φ_i $(i = 1, \ldots, n)$ that *approximates* $\mathbf{Y}$ as good as possible.

- **Solution:** Let $\{\Phi_i\}_{i=1}^n$ span a subspace $D_n \subset \mathbb{R}^n$, then the best approximation $\mathbf{Y}_{\hat{\theta}}$ to $\mathbf{Y}$ in D_n is the orthogonal projection.



- Define the inner product of vectors $x, y \in \mathbb{R}^N$ defined as

$$< x, y >= x^T y$$

- Orthogonal Projection: the approximation error $\mathbf{Y} - \mathbf{Y}_{\hat{\theta}}$ is orthogonal to Φ_i, or

$$< \Phi_i, (\mathbf{Y} - \mathbf{Y}_{\hat{\theta}}) >= \Phi_i^T (\mathbf{Y} - \mathbf{Y}_{\hat{\theta}}) = 0$$

  for all $i = 1, \ldots, n$.

- Consequently

$$\Phi^T (\mathbf{Y} - \mathbf{Y}_{\hat{\theta}}) = 0$$

- If $\hat{\theta}_{OP}$ orthogonal projection, then

$$\Phi^T (\mathbf{Y} - \mathbf{Y}_{\hat{\theta}_{OP}}) = 0 \Rightarrow (\Phi^T \Phi) \hat{\theta}_{OP} = \Phi^T \mathbf{Y}$$

- Using the weighted inner-product $< x, y >= x^T \mathbf{W} y$ yields WLS.

Least Squares, Average and Maximum Likelihood

- Simple model.
$$y(t) = \mu + e(t)$$
OLS:
$$\hat{\theta}_{LS} = \frac{1}{N}\sum_{t=1}^{N} y(t) = \bar{y}$$

- Statistical model.
$$y \sim \mathcal{N}(\mu, \lambda^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\lambda^2}\right)$$
Then the Likelihood function of $Y = x$ becomes
$$L_\mu(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\lambda^2}\right)$$

- If N independent datasamples $\{y(t)\}_t$, then

$$\hat{\theta}_{ML} = \max_{\mu} \prod_{t=1}^{N} L_{\mu}(y(\theta)) = \prod_{k=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)}{2\lambda^2}\right)$$

- Equivalently

$$\hat{\theta}_{ML} = \operatorname*{argmax}_{\mu} \log \prod_{t=1}^{N} L_{\mu}(y(t)) = \operatorname*{argmin}_{\mu} \sum_{t=1}^{N} (y(t)-\mu)^2$$

- So $\hat{\theta}_{ML} = \hat{\theta}_{LS} = \bar{\theta}$

Least Squares (Statistical Approach)

In order to explore properties of the least squares estimate we need to specify the system, i.e. we need to make assumptions on how the data was generated.

Assumptions:

- $\varphi(t)$ is known and deterministic. (Quite restrictive!)
- System
$$y(t) = \varphi(t)\theta_0 + e(t)$$
where $\theta_O \in \mathbb{R}^n$ is the *true* parameter, and $(e(t))_t$ is a sequence of random variables, with $E[e(t)] = 0$, $E[e(t)e(s)] = R_{ts}$, or
$$E[\mathbf{e}] = 0_n,\ E[\mathbf{ee}] = \mathrm{cov}(\mathbf{e}) = \mathbf{R} \in \mathbb{R}^{n\times n}$$

Rem. If $\mathbf{R} = \sigma^2 I_n$, then $\mathbf{e}$ is white noise with variance σ^2.

Least Squares (Statistical Approach), Ct'd

- The WLS is *unbiased* or $E\left[\hat{\theta}_{WLS}\right] = \theta_0$ when $E[\Phi^T \mathbf{e}] = 0_n$

$$(\Phi^T\Phi)^{-1}\Phi^T E[\Phi\theta_0 + e] = \theta_0$$

- Covariance matrix

$$\operatorname{cov}(\hat{\theta}) = E\left[(\hat{\theta} - E[\hat{\theta}])(\hat{\theta} - E[\hat{\theta}])^T\right]$$

as

$$\begin{aligned}
&(\hat{\theta}_{WLS} - E[\hat{\theta}_{WLS}]) \\
&\quad = (\Phi^T\mathbf{W}\Phi)^{-1}\Phi^T\mathbf{W}(\Phi\theta_0 + \mathbf{e}) - (\Phi^T\mathbf{W}\Phi)^{-1}(\Phi^T\mathbf{W}\Phi)\theta_0 \\
&\quad = (\Phi^T\mathbf{W}\Phi)^{-1}\Phi^T\mathbf{W}\mathbf{e}
\end{aligned}$$

then

$$\operatorname{cov}(\hat{\theta}_{WLS}) = (\Phi^T\mathbf{W}\Phi)^{-1}(\Phi^T\mathbf{W}\mathbf{R}\mathbf{W}\Phi)(\Phi^T\mathbf{W}\Phi)^{-1}$$

For OLS and $\mathbb{R} = \sigma^2 I_N$, then

$$\mathrm{cov}(\hat{\theta}_{LS}) = \frac{\sigma^2}{N}\left(\frac{1}{N}\Phi^T\Phi\right)^{-1}$$

- If $e(t) \sim \mathcal{N}(0, \mathbf{R})$ and Φ deterministic, then

$$\hat{\theta}_{WLS} \sim \mathcal{N}\left(\theta_0, \mathrm{cov}(\hat{\theta}_{WLS})\right).$$

Least Squares (optimal cost $V_N(\hat\theta_{LS})$)

Property

$$E[2V_N(\hat\theta_{LS})] = \sigma^2(N-n)$$

Proof

$$\begin{aligned} 2V_N(\hat\theta_{LS}) &= (Y-\Phi\hat\theta_{LS})^T(Y-\Phi\hat\theta_{LS}) \\ &= Y^TY - Y^T\Phi(\Phi^T\Phi)^{-1}\Phi^TY - Y^T\Phi(\Phi^T\Phi)^{-1}\Phi^TY \\ &\quad + Y^T\Phi(\Phi^T\Phi)^{-1}(\Phi^T\Phi)(\Phi^T\Phi)^{-1}Y \\ &= Y^T\left(I_N - \Phi(\Phi^T\Phi)^{-1}\Phi^T\right)Y \end{aligned}$$

Taking expectation gives

$$\begin{aligned} &E[2V_N(\hat\theta_{LS})] \\ &= E[(\Phi\theta_0+\mathbf{e})^T\left(I_N - \Phi(\Phi^T\Phi)^{-1}\Phi^T\right)(\Phi\theta_0+\mathbf{e})] \\ &= E[\mathbf{e}^T\left(I_N - \Phi(\Phi^T\Phi)^{-1}\Phi^T\right)\mathbf{e}]. \end{aligned}$$

Using the property that $\mathbf{e}^T A\mathbf{e} = \sum_{i=1}^n \sum_{j=1}^n (e_i e_j) A_{ij} = \operatorname{tr}(A(\mathbf{e}\mathbf{e}^T))$ for any matrix $A \in \mathbb{R}^{n\times n}$ and vector $\mathbf{e} \in \mathbb{R}^n$:

$$\begin{aligned} E[2V_N(\hat{\theta}_{LS})] &= \operatorname{tr}\left(\left(I_N - \Phi(\Phi^T\Phi)^{-1}\Phi^T\right) E[\mathbf{e}\mathbf{e}^T]\right) \\ &= \sigma^2 \left(\operatorname{tr}(I_N) - \operatorname{tr}(\Phi(\Phi^T\Phi)^{-1}\Phi^T)\right) \\ &= \sigma^2 \left(N - \operatorname{tr}((\Phi^T\Phi)^{-1}(\Phi^T\Phi)\right) \\ &= \sigma^2(N-n) \end{aligned}$$

where we also use linearity of tr, i.e. $\operatorname{tr}(AB) = \operatorname{tr}(BA)$ for arbitrary matrices. So $\hat{\sigma}^2 = \frac{2V_N(\hat{\theta}_{LS})}{N-n}$ is an unbiased estimate of σ^2.

Note that

$$E[2V_n(\theta_0)] = \mathbf{e}^T\mathbf{e} = \sigma^2 N$$

Least Squares (Statistical Approach), Ct'd

- An estimate $\hat{\theta}_1$ is more efficient than $\hat{\theta}_2$ if

$$\operatorname{cov}(\hat{\theta}_1) \leq \operatorname{cov}(\hat{\theta}_2)$$

- Which choice of $\mathbf{W}$ will yield a maximally efficient estimate $\hat{\theta}_{WLS}$?
- The choice $\mathbf{W} = \mathbf{R}^{-1}$ (if exists) yields optimal efficiency, or
  - $\hat{\theta}_{WLS} = \left(\Phi^T \mathbf{R}^{-1} \Phi\right)^{-1} \Phi^T \mathbf{R}^{-1} \mathbf{Y}$
  - $\operatorname{cov}(\hat{\theta}_{WLS}) = \left(\Phi^T \mathbf{R}^{-1} \Phi\right)^{-1}$

In this case the estimator is known as the BLUE (Best Linear Unbiased Estimate) or the Gauss-Markov estimate.

Least Squares (Statistical Approach), Ct'd

- BLUE $=$ Best Linear Unbiased Estimate.
- When $\mathbf{e}$ white noise ($\mathbf{R} = \sigma^2 I_n$), BLUE=OLS.
- If $\mathbf{e}$ Gaussian, BLUE is best possible estimator. If $\mathbf{e}$ non-Gaussian, there might exist better 'non-linear' estimators.
- BLUE can be derived also for non-invertible covariance matrices $\mathbf{R}$.

Least Squares (Computational Aspects)

The OLS solution is unsuitable for direct numerical computation. Alternatives avoid computation of the inverse $(\Phi^T\Phi)^{-1}$

- Use the *normal equations* instead:

$$(\Phi^T\Phi)\hat{\theta}_{LS} = \Phi^T\mathbf{Y}$$

- Solve an *overdetermined system of linear equations*

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

  Stable numerical procedures:

  – QR-Factorization
  – SVD-Factorization
  – Using Pseudo-inverse

Least Squares (Computational Aspects)

QR-Factorization Let $\Phi = \mathbf{QR}$ where $\mathbf{Q} \in \mathbb{R}^{N\times N}$ orthonormal ($\mathbf{QQ}^T = \mathbf{Q}^T\mathbf{Q} = I_N$) and $\mathbb{R}^{N\times n}$ upper triangular, then instead of solving

$$\mathbf{Y} \approx \Phi\theta$$

one may equivalently solve

$$\mathbf{Q}^T\mathbf{Y} = \mathbf{Q}^T\Phi\theta = \mathbf{R}\theta$$

which is easy due to the structure of $\mathbf{R}$:

- Requires more computation than solving the normal equations.
- Less sensitive to rounding errors

Least Squares (Computational Aspects), Ct'd

SVD: Let

$$\Phi = \mathbf{U}^T \Sigma \mathbf{V}$$

with $\mathbf{U} \in \mathbb{R}^{N\times N}, \mathbf{V} \in \mathbb{R}^{n\times n}$, $\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = I_N, \mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = I_n$

$$\Sigma = \begin{bmatrix} \mathrm{diag}(\sigma_1, \ldots, \sigma_n) \\ 0 \end{bmatrix} \in \mathbb{R}^{N\times n}$$

containing the singular values. Then $(\Phi^T\Phi)^{-1} = \mathbf{V}^T\Sigma^{-2}\mathbf{V}$ and

$$\hat{\theta}_{LS} = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{Y}$$

$$= (\mathbf{V}^T\Sigma^{-2}\mathbf{V})(\mathbf{V}^T\Sigma\mathbf{U})\mathbf{Y} = \mathbf{V}^T \begin{bmatrix} \mathrm{diag}(\sigma_1^{-1}, \ldots, \sigma_n^{-1}) \\ 0 \end{bmatrix}^T \mathbf{U}\mathbf{Y}$$

In MATLAB:

```
>> [U,S,V] = svd(Phi);
>> theta = V*pinv(S)*U*Y
```

Least Squares (Computational Aspects), Ct'd

Pseudo-inverse Defined as

$$\Phi^{\dagger} = \mathbf{U}^T \Sigma^{\dagger} \mathbf{V}$$

and

$$(\Phi^T \Phi)^{\dagger} = \mathbf{V}^T \Sigma^{\dagger} \Sigma^{\dagger} \mathbf{V}$$

Then

$$\hat{\theta}_{LS^{\dagger}} = (\Phi^T \Phi)^{\dagger} \Phi^T \mathbf{Y}$$
$$= (\mathbf{V}^T \Sigma^{\dagger} \Sigma^{\dagger} \mathbf{V})(\mathbf{V}^T \Sigma \mathbf{U})\mathbf{Y} = \mathbf{V}^T \Sigma^{\dagger} \mathbf{U} \mathbf{Y}$$

- Avoids singularity issues.
- If multiple solutions possible, take lowest $\|\theta\|_2$

In MATLAB:

```
>> theta = pinv(Phi)*Y
```

Conclusions

- Regression (linear in the parameters) models describe a large class of dynamical models.
- The LS estimator is fundamental in SI and can be derived from various perspectives.
- We have assumed that Φ is deterministic. We run into problems when this matrix is a function of stochastic variables (ARX).