3F3 - Detection, Estimation and Inference for Signal Processing
Part II - Estimation Theory and Inference

Created by S.J. Godsill
Edited by S.S. Singh

Signal Processing and Communications Laboratory,
Engineering Department,
Cambridge, UK

Estimation Theory and Inference Methods I

- We have already solved an *estimation problem* in Wiener filtering. In general, Statistical analysis is an imprecise umbrella term that covers the problem of designing a statistical models for data sets, fitting the models to the data sets, assessing the goodness of the fit, making inferences about the unobserved variables of the model, predicting future trends and values.
- Simple examples include the estimation of mean and variance for a collection of random measurements, while more sophisticated examples might involve the estimation of parameters for some highly complex probability model of a signal, such as a multiple sinusoid model or an autoregressive model.

Estimation Theory and Inference Methods II

- We will first look at a general linear modelling framework for signals and data. We will then consider estimation and inference techniques for its analysis. A more precise distinction between estimation and inference will subsequently be given.

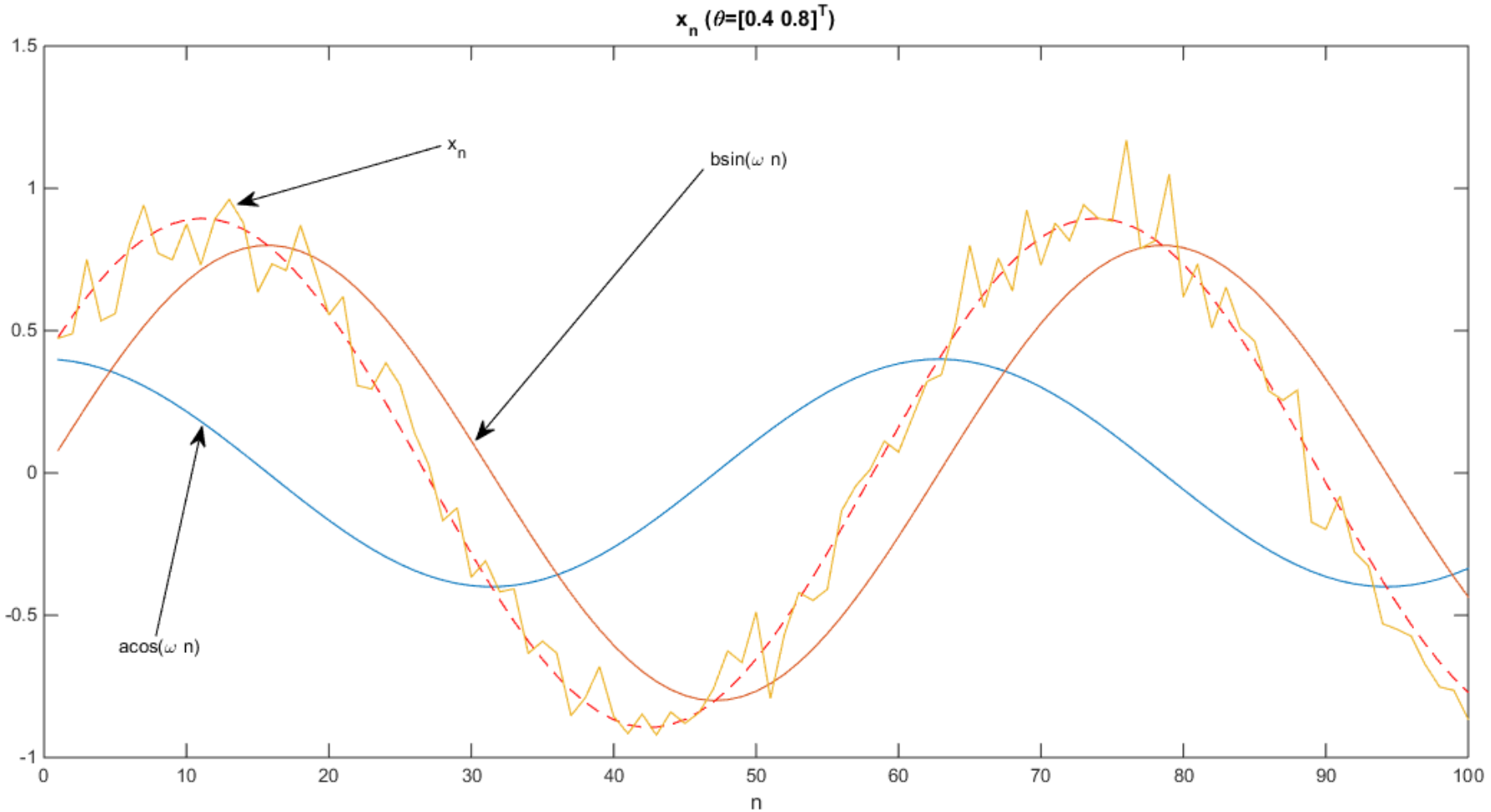
Example: the Sinusoidal Model I

- This model forms the basic building block for many frequency and spectral estimation algorithms. It is also used in sinusoidal speech coders.
- The figure shows data points $x_0, x_1, x_2, \ldots$ that arise from a noisy sinusoidal model.
- The following model expresses the data as the sum of a sine and a cosine component at a particular (assumed known for now) frequency ω corrupted by noise e_n:

$$x_n = a\cos(\omega n) + b\sin(\omega n) + e_n$$

- Parameters a and b are unknown. [This model is equivalent to a sinusoid with uniformly random phase and random amplitude, see examples paper for justification...]

Example: the Sinusoidal Model II



Example: the Sinusoidal Model III

Example Matlab code and figure for generating from the sinusoidal model:

```
omega=0.1;
N=100;
sigma_e=0.1;

G=[cos((1:N)*omega)' sin((1:N)*omega)'];

theta=[0.4;0.8];

x=randn(N,1)*sigma_e+G*theta;

figure

plot(1:N,G(:,1)*theta(1),1:N,G(:,2)*theta(2),1:N,G*theta,'r--',1:N,x)
title('x_n (\theta=[0.4 0.8]^T)')
xlabel('n')
```

Example: the Sinusoidal Model IV

- We can express this setup in a standard format that will encompass many other variations of the model. First vectorise the data as

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix}$$

- Let

$$\mathbf{G} = \begin{bmatrix} \cos(0) & \sin(0) \\ \cos(\omega) & \sin(\omega) \\ \vdots & \vdots \\ \cos((N-1)\omega) & \sin((N-1)\omega) \end{bmatrix} = \begin{bmatrix} \mathbf{c}(\omega) & \mathbf{s}(\omega) \end{bmatrix}$$

and

$$\boldsymbol{\theta} = \begin{bmatrix} a \\ b \end{bmatrix}$$

Example: the Sinusoidal Model V

- The expression may be written for the whole vector $\mathbf{x}$ as

$$\boxed{\mathbf{x} = \mathbf{G}\boldsymbol{\theta} + \mathbf{e}}$$

  where $\mathbf{e} = \left[e_0, e_1, \ldots, e_{N-1}\right]^{\mathrm{T}}$
- Choice of the matrix $\mathbf{G}$ and vector of *unknowns* $\boldsymbol{\theta}$ will results in different models.
- We have just described the *General Linear Model (GLM.)* The matrix $\mathbf{G}$ is known as the *Design matrix* or the *Regression matrix*. As we will see, choosing $\mathbf{G}$ and model's parameters $\boldsymbol{\theta}$ differently will lead to different models for different applications.

Example: the Sinusoidal Model VI

- For example, we can build a much more complex model composed of J sinusoids at different frequencies ω_j:

$$x_n = \sum_{j=1}^{J} a_j \cos(\omega_j n) + b_j \sin(\omega_j n) + e_n$$

and the linear model expression is still $\mathbf{x} = \mathbf{G}\boldsymbol{\theta} + \mathbf{e}$

$$\mathbf{G} = \begin{bmatrix} \mathbf{c}(\omega_1) & \mathbf{s}(\omega_1) & \mathbf{c}(\omega_2) & \mathbf{s}(\omega_2) & \dots & \mathbf{c}(\omega_J) & \mathbf{s}(\omega_J) \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} a_1, b_1, a_2, b_2, \dots a_J, b_J \end{bmatrix}^{\mathrm{T}}$$

Example: the Sinusoidal Model VII

- Thus we could make up a very complicated signal composed of lots of 'sinusoids' (with known frequencies) all added together. If we estimate the parameters $\boldsymbol{\theta}$ from some data then we will be doing a kind of probabilistic 'spectrum estimation' - see e.g. Data Analysis project SF1.

Least squares estimation of the General Linear Model I

The general linear model has the following form:

$$\mathbf{x} = \mathbf{G}\boldsymbol{\theta} + \mathbf{e}$$

First let us derive the *Ordinary Least Squares* estimator of $\boldsymbol{\theta}$ for the general linear model, a formula that should be familiar from 1B. Here we will carry it out using matrix-vector derivatives

- We attempt to find the 'best fit' model that matches the data by minimising the following error:

$$J(\boldsymbol{\theta}) = \sum_{n=0}^{N-1} e_n^2 = \mathbf{e}^T\mathbf{e}$$

- Expand this using $\mathbf{e} = \mathbf{x} - \mathbf{G}\boldsymbol{\theta}$,

$$\mathbf{e}^T\mathbf{e} = (\mathbf{x} - \mathbf{G}\boldsymbol{\theta})^T(\mathbf{x} - \mathbf{G}\boldsymbol{\theta}) = \mathbf{x}^T\mathbf{x} + \boldsymbol{\theta}^T\mathbf{G}^T\mathbf{G}\boldsymbol{\theta} - 2\boldsymbol{\theta}^T\mathbf{G}^T\mathbf{x}$$

Least squares estimation of the General Linear Model II

- Now, defining the vector *gradient* in the usual way:

$$\nabla J = \begin{bmatrix} \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_0} \\ \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_1} \\ \vdots \\ \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_{P-1}} \end{bmatrix}$$

we obtain:

$$\nabla J = 2\mathbf{G}^T\mathbf{G}\boldsymbol{\theta} - 2\mathbf{G}^T\mathbf{x}$$

and finally, at a stationary point, setting $\nabla J = \mathbf{0}$,

$$\mathbf{G}^T\mathbf{G}\boldsymbol{\theta} = \mathbf{G}^T\mathbf{x}$$

or, for invertible $\mathbf{G}^T\mathbf{G}$,

$$\boldsymbol{\theta}^{OLS} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{x}$$

i.e. the classical *Ordinary Least Squares* estimate of $\boldsymbol{\theta}$.

Least squares estimation of the General Linear Model III

- Another useful way to think about the expansion of J is by 'completing the square':

$$\mathbf{x}^T\mathbf{x}+\boldsymbol{\theta}^T\mathbf{G}^T\mathbf{G}\boldsymbol{\theta}-2\boldsymbol{\theta}^T\mathbf{G}^T\mathbf{x}$$
$$=(\boldsymbol{\theta}-\boldsymbol{\theta}^{OLS})^T\mathbf{G}^T\mathbf{G}(\boldsymbol{\theta}-\boldsymbol{\theta}^{OLS})-\boldsymbol{\theta}^{OLS\,T}\mathbf{G}^T\mathbf{x}+\mathbf{x}^T\mathbf{x}$$

  This serves to show that the OLS estimator is globally optimal, and will also come in handy shortly under likelihood and Bayesian inference schemes.
- We will come back to this under Maximum Likelihood estimation, but for now consider the properties of the OLS estimator for the General Linear Model.

Bias of the Least Squares Estimator I

- The OLS estimator of the General Linear Model is a *linear estimator*, since it is a matrix multiplied by the data $\mathbf{x}$.
- What are its properties, and could we ever do better than OLS?
- First consider the *bias*:

$$\mathbb{E}[\boldsymbol{\theta}^{OLS}] = \mathbb{E}[(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{x}] = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbb{E}[\mathbf{x}]$$

- But, for the linear model, $\mathbf{x} = \mathbf{G}\boldsymbol{\theta} + \mathbf{e}$, so

$$\mathbb{E}[\mathbf{x}] = \mathbf{G}\boldsymbol{\theta} + \mathbf{0} = \mathbf{G}\boldsymbol{\theta}$$

since the noise process $\{e_n\}$ has zero mean.

- Substituting back into the first expectation gives

$$\mathbb{E}[\boldsymbol{\theta}^{OLS}] = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{G}\boldsymbol{\theta} = (\mathbf{G}^T\mathbf{G})^{-1}(\mathbf{G}^T\mathbf{G})\boldsymbol{\theta} = \boldsymbol{\theta}$$

hence proving that OLS is *unbiased*, which sounds like good news.

Covariance of OLS I

- How about its variance compared to other linear estimators? Define the OLS matrix term as

$$\mathbf{C} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T$$

- Then examine the variance of any other *unbiased* estimator, which we write as:

$$\hat{\boldsymbol{\theta}} = \mathbf{D}\mathbf{x}$$

where

$$\mathbf{D} = \mathbf{C} + \boldsymbol{\Delta}$$

and $\boldsymbol{\Delta}$ is some matrix perturbation away from the OLS solution.

Covariance of OLS II

- For **Dx** to be unbiased we require as above that

$$\mathbb{E}[\mathbf{Dx}] = \boldsymbol{\theta},$$

i.e.

$$\mathbb{E}[(\mathbf{C}+\boldsymbol{\Delta})\mathbf{x}] = (\mathbf{C}+\boldsymbol{\Delta})\mathbb{E}[\mathbf{x}] = (\mathbf{C}+\boldsymbol{\Delta})\mathbf{G}\boldsymbol{\theta} = \boldsymbol{\theta}+\boldsymbol{\Delta}\mathbf{G}\boldsymbol{\theta} = \boldsymbol{\theta},$$

and therefore we require

$$\boldsymbol{\Delta}\mathbf{G} = \mathbf{0}$$

since it must work for *all* $\boldsymbol{\theta}$

Covariance of OLS III

- Now, the covariance matrix of the estimator of $\boldsymbol{\theta}$ is

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

  This is the matrix-vector version of the scalar variance of a random variable. Note in particular that the (i, i)th element of the covariance matrix is the *variance* of $\hat{\theta}_{i-1}$.

- But since we are dealing only with unbiased estimators we have $\mathbb{E}[\hat{\boldsymbol{\theta}}] = \boldsymbol{\theta}$ and the calculation reduces to the covariance matrix of the estimation error:

$$\text{cov}(\hat{\boldsymbol{\theta}}) = \mathbb{E}[(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T] = \mathbb{E}[\hat{\boldsymbol{\theta}}\hat{\boldsymbol{\theta}}^T] - \boldsymbol{\theta}\boldsymbol{\theta}^T$$

- Now, for the linear estimator $\hat{\boldsymbol{\theta}} = \mathbf{Dx}$, we have

$$\mathbb{E}[\hat{\boldsymbol{\theta}}\hat{\boldsymbol{\theta}}^T] = \mathbb{E}[\mathbf{D}\mathbf{x}\mathbf{x}^T\mathbf{D}^T] = \mathbf{D}\mathbb{E}[\mathbf{x}\mathbf{x}^T]\mathbf{D}^T$$

Covariance of OLS IV

- But,

$$\mathbf{x}\mathbf{x}^T = (\mathbf{G}\boldsymbol{\theta}+\mathbf{e})(\mathbf{G}\boldsymbol{\theta}+\mathbf{e})^T = \mathbf{G}\boldsymbol{\theta}\boldsymbol{\theta}^T\mathbf{G}^T + \mathbf{e}\mathbf{e}^T + \mathbf{e}\boldsymbol{\theta}^T\mathbf{G}^T + \mathbf{G}\boldsymbol{\theta}\mathbf{e}^T$$

and hence

$$\mathbb{E}[\mathbf{x}\mathbf{x}^T] = \mathbf{G}\boldsymbol{\theta}\boldsymbol{\theta}^T\mathbf{G}^T + \sigma_e^2\mathbf{I}$$

since $\{e_n\}$ is zero mean white noise with variance σ_e^2.

- Now the expectation is obtained as

$$\begin{aligned}\mathbb{E}[\hat{\boldsymbol{\theta}}\hat{\boldsymbol{\theta}}^T] &= \mathbf{D}\mathbb{E}[\mathbf{x}\mathbf{x}^T]\mathbf{D}^T \\ &= \mathbf{D}(\mathbf{G}\boldsymbol{\theta}\boldsymbol{\theta}^T\mathbf{G}^T + \sigma_e^2\mathbf{I})\mathbf{D}^T \\ &= \boldsymbol{\theta}\boldsymbol{\theta}^T + \sigma_e^2\mathbf{D}\mathbf{D}^T\end{aligned}$$

[this last line is obtained by substituting for $\mathbf{D} = \mathbf{C} + \boldsymbol{\Delta}$ and using the unbiasedness criterion, $\boldsymbol{\Delta}\mathbf{G} = \mathbf{0}$].

Covariance of OLS V

- Then we have,

$$\begin{aligned}
\text{cov}(\hat{\boldsymbol{\theta}}) &= \mathbb{E}[\hat{\boldsymbol{\theta}}\hat{\boldsymbol{\theta}}^T] - \boldsymbol{\theta}\boldsymbol{\theta}^T = \sigma_e^2 \mathbf{D}\mathbf{D}^T \\
&= \sigma_e^2 (\mathbf{C} + \boldsymbol{\Delta})(\mathbf{C} + \boldsymbol{\Delta})^T \\
&= \sigma_e^2 (\mathbf{C}\mathbf{C}^T + \boldsymbol{\Delta}\boldsymbol{\Delta}^T + \boldsymbol{\Delta}\mathbf{C}^T + \mathbf{C}\boldsymbol{\Delta}^T) \\
&= \sigma_e^2 (\mathbf{C}\mathbf{C}^T + \boldsymbol{\Delta}\boldsymbol{\Delta}^T) \\
&\qquad [\text{since } \boldsymbol{\Delta}\mathbf{C}^T = \mathbf{C}\boldsymbol{\Delta}^T = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\boldsymbol{\Delta}^T = \mathbf{0}))] \\
&= \sigma_e^2 ((\mathbf{G}^T\mathbf{G})^{-1} + \boldsymbol{\Delta}\boldsymbol{\Delta}^T)
\end{aligned}$$

- Clearly with $\boldsymbol{\Delta} = \mathbf{0}$ we have the OLS estimator covariance, so

$$\text{cov}(\hat{\boldsymbol{\theta}}) = \text{cov}(\boldsymbol{\theta}^{OLS}) + \sigma_e^2 \boldsymbol{\Delta}\boldsymbol{\Delta}^T$$

Covariance of OLS VI

- Now, the variance of each parameter estimate $\hat{\theta}_{i-1}$ is the ith diagonal element of $\text{cov}(\hat{\boldsymbol{\theta}})$. And the diagonal elements of $\boldsymbol{\Delta}\boldsymbol{\Delta}^T$ are also ≥ 0 by its construction. Hence we have that

$$\text{var}(\hat{\theta}_i) \geq \text{var}(\theta_i^{OLS})$$

  for each parameter $i = 0, ..., P-1$, with equality being achieved when $\boldsymbol{\Delta} = 0$, corresponding to the OLS estimator.
- We have thus proved that the OLS estimator is the minimum variance unbiased estimator of $\boldsymbol{\theta}$. Such an estimator is termed a Best Linear Unbiased Estimator (BLUE)
- It is fairly straightforward to show that the OLS is the unique BLUE for the General Linear Model.

Covariance of OLS VII

- To prove these results for the OLS method all we needed to assume about the model was that the noise process $\{e_n\}$ is zero-mean and white - no probability distribution needed to be assumed for $\{e_n\}$.
- However, if we do know the distribution of $\{e_n\}$, e.g. Gaussian white noise, or some other distribution, then it could be possible that the linear estimator can be beaten by a *nonlinear* estimation method [in fact it turns out that in the white Gaussian noise case, OLS is the global best unbiased estimator; not so however for correlated Gaussian noise, or non-Gaussian noise].
- If in addition we have prior probability information $p(\boldsymbol{\theta})$ about $\boldsymbol{\theta}$ then a Bayesian estimator can give better mean-squared error performance at the cost of some small bias in the estimates - see later

Another example of a GLM: Autoregressive (AR) model I

- The AR model is a standard time series model based on an all-pole filter applied to white noise:

$$x_n = \sum_{i=1}^{P} a_i x_{n-i} + e_n. \tag{1}$$

  where e_n is zero mean white noise with variance σ_e^2
- The coefficients $\{a_i : i = 1, \ldots, P\}$ are the filter coefficients of the all-pole filter, the AR parameters, and P, the number of coefficients, is the order of the AR process.
- $\{e_n\}$ can be interpreted as a 'prediction error' when predicting the next data point from the previous P.

Another example of a GLM: Autoregressive (AR) model II

- Recall that the transfer function for the filter is:

$$H(z) = \frac{1}{1 - \sum_{i=1}^{P} a_i z^{-i}}$$

- And hence the power spectrum for the model is:

$$\mathcal{S}_x(e^{j\omega}) = |H(\exp(j\omega))|^2 \sigma_e^2 = \frac{\sigma_e^2}{|1 - \sum_{i=1}^{P} a_i e^{-j\omega i}|^2}$$

- The shape of the power spectrum may readily be sketched by first sketching the magnitude frequency response of $H(z)$ and then squaring
- The model is used extensively in linear prediction of speech, speech synthesis and coding (especially in its adaptation to low bitrate CELP encoders).

Another example of a GLM: Autoregressive (AR) model III

- The AR model can be expressed as in equation (1). For N data samples $x_0, x_1, \ldots, x_{N-1}$

$$\mathbf{x} = \mathbf{G}\,\mathbf{a} + \mathbf{e} \tag{2}$$

where $\mathbf{e}$ is the vector of N error values and the $(N \times P)$ matrix $\mathbf{G}$ is given by

$$\mathbf{G} = \begin{bmatrix} x_{-1} & x_{-2} & \cdots & x_{-(P-1)} & x_{-P} \\ x_0 & x_{-1} & \cdots & x_{-(P-2)} & x_{-(P-1)} \\ \vdots & & \ddots & & \vdots \\ x_{N-2} & x_{N-3} & \cdots & x_{N-P} & x_{N-P-1} \end{bmatrix} \tag{3}$$

- Note that the matrix $\mathbf{G}$ contains data values prior to time $n = 0$ in order to calculate all the error terms $e_n,\ n = 0, ..., N-1$.

Another example of a GLM: Autoregressive (AR) model IV

- In practise, if N is much larger than P you can assume $x_n = 0$ for $n < 0$ when setting up the generalised linear model equation for the block of data $x_0, x_1, \ldots, x_{N-1}$.
- Alternatively, you can write the general linear model for the reduced data vector $\mathbf{x} = [x_P, x_{P+1}, \ldots, x_N]^{\mathrm{T}}$. When $N >> P$ either way is fine when solving for the model parameter $\boldsymbol{\theta} = \mathbf{a}$.
- Note we can't directly generate a vector of data from an AR model using the formula $\mathbf{x} = \mathbf{G}\boldsymbol{\theta} + \mathbf{e}$, where $\mathbf{e}$ is generated from some suitable noise process, such as white Gaussian noise, since $\mathbf{G}$ now depends on the data we are trying to generate.
- However, it is straightforward to generate the data sequentially by applying the all-pole IIR filter with coefficients $a_1, \ldots, a_P$ to a white noise input signal, see code example below.

Another example of a GLM: Autoregressive (AR) model V

- Alternatively, we can generate data using the following method: assume $x_n = 0$ for $n < 0$. Let $e_0, e_1, \ldots$ be independent Gaussian random variables of mean 0 and variance σ_e^2. Now execute equation (1) until $n = N + B$. Discard $x_0, x_1, \ldots, x_B$ and keep the remaining N samples. Choose variable B so that $B >> P$. B is the "burn-in" time, the time we wait before collecting valid AR samples, to ensure the AR model has "forgotten" its (incorrect) initialisation.

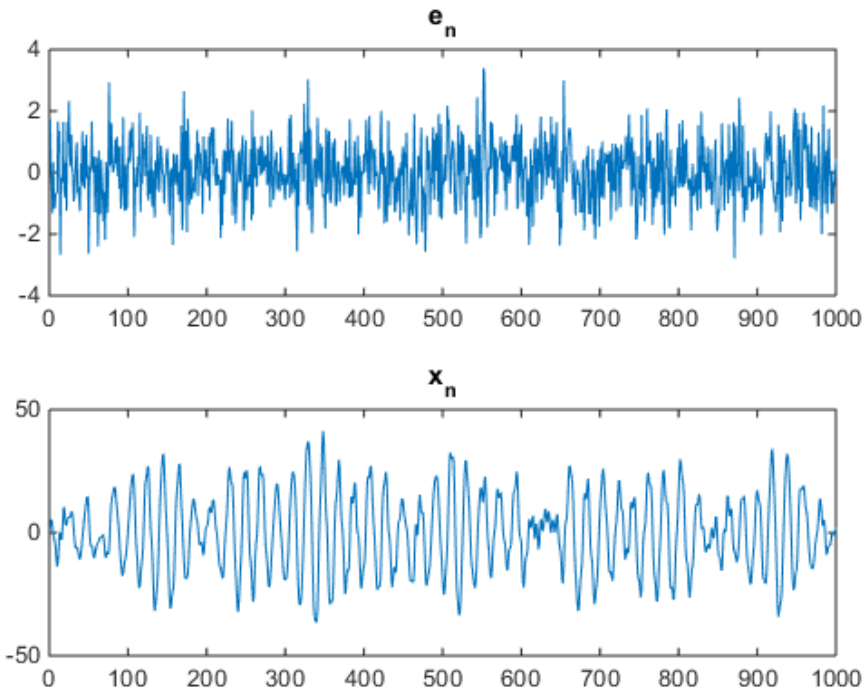
Another example of a GLM: Autoregressive (AR) model VI

- Code and plot for simulation of AR model data:

```
N=1000;
P=4;
pole(1)=0.99*exp(j*0.1*pi);
pole(3)=conj(pole(1));
pole(2)=0.97*exp(j*0.4*pi);
pole(4)=conj(pole(2));
a=poly(pole)
sigma_e=1;
e=randn(N,1)*sigma_e;
% Past data points assumed to be zero:
x=filter(1,a,e);
figure
subplot(211), plot(e)
title('e_n')
subplot(212), plot(x)
title('x_n')
% Make a Matlab system model:
sys=tf(1,a,1);
figure, pzplot(sys)
title('Poles of AR(4) model')
[H,w]=freqz(1,a);
figure
subplot(211),
semilogy(w,abs(H).^2*sigma_e^2)
title('Power spectrum of AR(4) model')
subplot(212),
X=(abs(fft(x)).^2);
semilogy((0:N/2-1)*pi*2/N,X(1:N/2))
title('|DFT|^2 of x_n')
```

Another example of a GLM: Autoregressive (AR) model VII

AR excitation signal e_n



AR signal x_n

Figure 1 : Autoregressive data with $P = 4$, poles at $(r, \theta) = 0.99 \exp(\pm j0.1\pi)$ and $(r, \theta) = 0.97 \exp(\pm j0.4\pi)$

Example: OLS for the AR model I

- We can apply the OLS method directly to the AR model.
- We know for the AR model the form of the general linear model is:

$$\mathbf{x} = \mathbf{G}\,\mathbf{a} + \mathbf{e} \tag{4}$$

where $\mathbf{e}$ is the vector of N error values and the $(N \times P)$ matrix $\mathbf{G}$ is given by

$$\mathbf{G} = \begin{bmatrix} x_{-1} & x_{-2} & \cdots & x_{-(P-1)} & x_{-P} \\ x_0 & x_{-1} & \cdots & x_{-(P-2)} & x_{-(P-1)} \\ \vdots & & \ddots & & \vdots \\ x_{N-2} & x_{N-3} & \cdots & x_{N-P} & x_{N-P-1} \end{bmatrix} \tag{5}$$

Example: OLS for the AR model II

- Then, measure some data $\mathbf{x}$, construct $\mathbf{G}$ as above from the data vector (including the P samples required before $n = 0$) and estimate the parameters by

$$\mathbf{a}^{OLS} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\,\mathbf{x}$$

- This method does not guarantee that estimated all-pole filter is *stable* - poles can be estimated outside the unit circle.
- View the code and results of estimation for our synthetic $P = 4$ data in the figures below - for $N = 100$ and $N = 500$, showing improved fitting with larger number of data points.

Example: OLS for the AR model III

```
N=500;
P=4;
pole(1)=0.99*exp(j*0.1*pi);
pole(3)=conj(pole(1));
pole(2)=0.97*exp(j*0.4*pi);
pole(4)=conj(pole(2));
a=poly(pole)
sigma_e=1;
e=randn(N,1)*sigma_e;
% Past data points assumed to be zero:
x=filter(1,a,e);
figure
subplot(211), plot(e)
title('e_n')
subplot(212), plot(x)
title('x_n')
% Make a Matlab system model:
sys=tf(1,a,1);
figure, pzplot(sys)
title('Poles of AR(4) model')
[H,w]=freqz(1,a);
figure
subplot(211),
semilogy(w,abs(H).^2*sigma_e^2)
title('Power spectrum of AR(4) model')
subplot(212),
X=(abs(fft(x)).^2);
semilogy((0:N/2-1)*pi*2/N,X(1:N/2))
title('|DFT|^2 of x_n')
```

Example: OLS for the AR model IV

```
N=length(x);

for q=1:P
   G(1:N-P,q)=x(P-q+1:N-q);
end
a_OLS=(G'*G)\G'*x(5:end)
a_OLS=[1; -a_OLS];

[H_OLS,w]=freqz(1,a_OLS);
figure
subplot(221),
plot(x),title('Signal x_n')
xlabel('n')
subplot(222),
semilogy(w,abs(H).^2*sigma_e^2,'--',w,abs(H_OLS).^2*sigma_e^2)
title('Power spectrum of AR(4) model')
legend('True','Estimated')

xlabel('\Omega')
subplot(223),
X=(abs(fft(x)).^2);
semilogy((0:N/2-1)*pi*2/N,X(1:N/2))
title('|DFT|^2 of x_n')
xlabel('\Omega')

sys_OLS=tf(1,a_OLS',1);
subplot(224), pzplot(sys,sys_OLS)
title('Poles of AR(4) model Blue =true, Red=estimated')
title('Poles of AR(4) model')
legend('True poles','Estimated poles')
```

Example: OLS for the AR model V

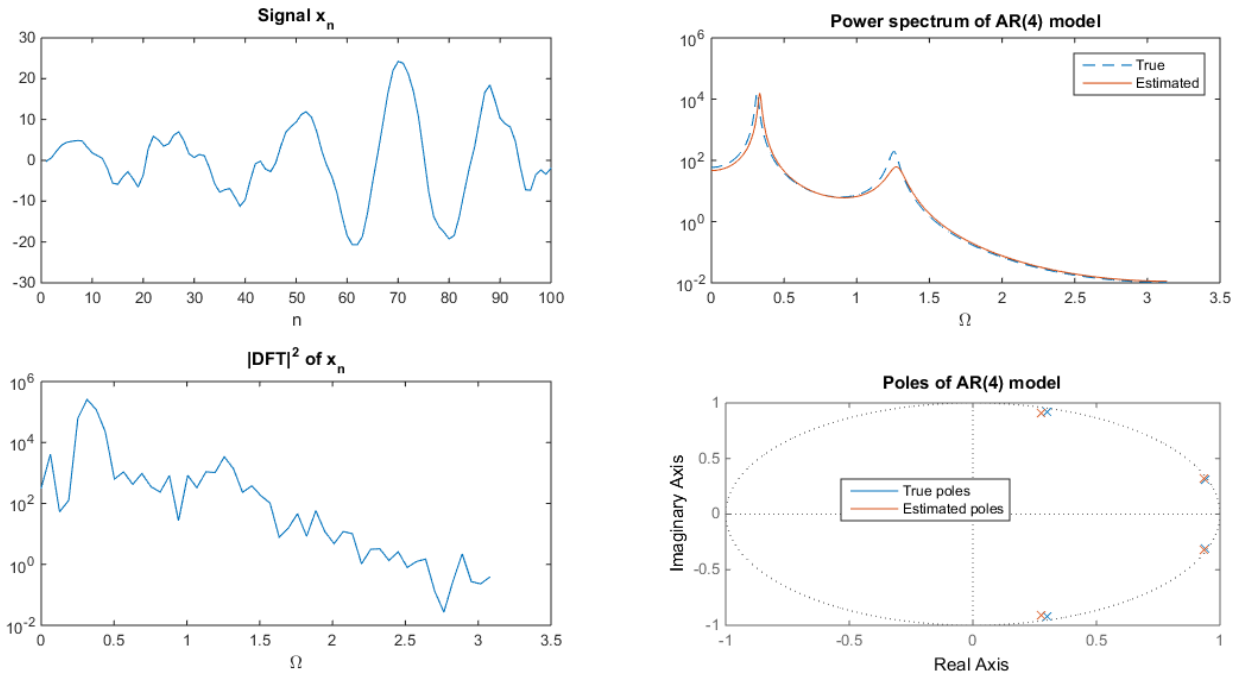


Figure 2 : Autoregressive data with $P = 4$, poles at $(r, \theta) = 0.99 \exp(\pm j 0.1\pi)$ and $(r, \theta) = 0.97 \exp(\pm j 0.4\pi)$, $N = 100$

Example: OLS for the AR model VI

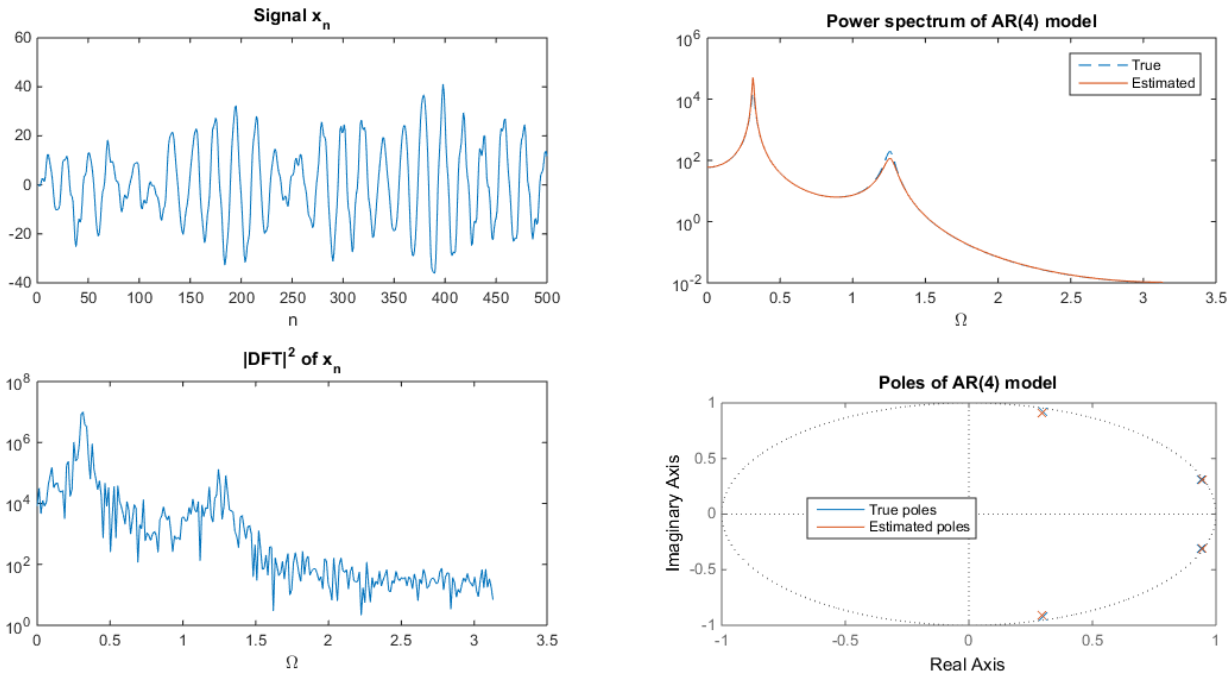


Figure 3 : Autoregressive data with $P = 4$, poles at $(r, \theta) = 0.99 \exp(\pm j0.1\pi)$ and $(r, \theta) = 0.97 \exp(\pm j0.4\pi)$, $N = 500$

Likelihood Estimation I

- The observed data $\mathbf{x} = \mathbf{G}\boldsymbol{\theta} + \mathbf{e}$ is random since $\mathbf{e}$ is a random vector.
- Knowing the p.d.f. of $\mathbf{e}$ and the values of $\mathbf{G}$ and $\boldsymbol{\theta}$, from the formula for the transformation of random variables we can derive the p.d.f. of $\mathbf{x}$, which is denoted as $p(\mathbf{x} \mid \boldsymbol{\theta})$.
- We then maximise $p(\mathbf{x} \mid \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ to find the best fitting model for the data.
- $p(\mathbf{x} \mid \boldsymbol{\theta})$ when $\mathbf{x}$ is fixed and treated as a function of $\boldsymbol{\theta}$ is clearly not a probability density function for $\boldsymbol{\theta}$.
- In Statistical language $p(\mathbf{x} \mid \boldsymbol{\theta})$ when regarded as a function of $\boldsymbol{\theta}$ is called the *Likelihood* function:

$$L(\boldsymbol{\theta} \mid \mathbf{x}) = p(\mathbf{x} \mid \boldsymbol{\theta}) \tag{6}$$

Likelihood Estimation II

- The Maximum Likelihood (ML) estimate for $\boldsymbol{\theta}$ is then that value of $\boldsymbol{\theta}$ which maximises the likelihood (for the given observation $\mathbf{x}$):

$$\boxed{\boldsymbol{\theta}^{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta} \mid \mathbf{x})} \tag{7}$$

Maximum likelihood (ML) estimator

- The value of $\boldsymbol{\theta}^{\mathrm{ML}}$ that maximises the p.d.f. in the right-hand side of (6) at the observed data $\mathbf{x}$ is clearly more likely to have been the $\boldsymbol{\theta}$ value that generated the data. This is the underlying rationale of ML estimation.

Likelihood Estimation III

- The maximisation task required for ML estimation can be achieved using standard differential calculus for well-behaved and differentiable likelihood functions, and it is often convenient analytically to maximise the log-likelihood function

$$l(\boldsymbol{\theta} \mid \mathbf{x}) = \log L(\boldsymbol{\theta} \mid \mathbf{x})$$

  rather than $L(\boldsymbol{\theta} \mid \mathbf{x})$ itself. Since log is a monotonically increasing function the two solutions are identical.
- In data analysis and signal processing applications the likelihood function is arrived at through knowledge of the stochastic model for the data.

Likelihood Estimation IV

- A convenient and versatile assumption for the noise in the GLM

$$\mathbf{x} = \mathbf{G}\boldsymbol{\theta} + \mathbf{e}$$

is that e_i are independent and identically distributed (iid) as zero-mean Gaussian variables:

$$p(\mathbf{e}) = \prod_{n=0}^{N-1} \mathcal{N}(e_n|0, \sigma_e^2) = \prod_{n=0}^{N-1} \frac{1}{\sqrt{2\pi\sigma_e^2}} e^{-\frac{1}{2\sigma_e^2} e_n^2}$$

- Notice in particular the sum-squared of error terms in the exponent: we will see in a moment that the ML solution is identical to the OLS solution for the linear Gaussian model.

Likelihood Estimation V

- We can write this in vector form as follows:

$$\prod_{n=0}^{N-1} \frac{1}{\sqrt{2\pi\sigma_e^2}} e^{-\frac{1}{2\sigma_e^2} e_n^2} = \frac{1}{\sqrt{2\pi\sigma_e^2}^N} e^{-\frac{1}{2\sigma_e^2} \sum_{n=0}^{N-1} e_n^2}$$
$$= \frac{1}{\sqrt{2\pi\sigma_e^2}^N} e^{-\frac{1}{2\sigma_e^2} \mathbf{e}^T \mathbf{e}}$$

  which we can recognise as the multivariate Gaussian distribution with mean zero and covariance matrix $\sigma_e^2 \mathbf{I}$:

$$p(\mathbf{e}) = \mathcal{N}(\mathbf{e}|\mathbf{0}, \sigma_e^2 \mathbf{I})$$

- Now, to get the p.d.f. $p(\mathbf{x} \mid \boldsymbol{\theta})$, notice that the linear model's equation $\mathbf{x} = \mathbf{G}\boldsymbol{\theta} + \mathbf{e}$ defines a random vector $\mathbf{x}$ from a random vector $\mathbf{e}$:

$$\mathbf{e} \rightarrow \mathbf{x}$$

Likelihood Estimation VI

- We can thus use the change of variables formula. Both $\boldsymbol{\theta}$ and $\mathbf{G}$ are constant terms in the change of variables.
- Hence the change of variables is a very simple one with unity Jacobian and we get

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = p(\mathbf{e})\big|_{\mathbf{e}=\mathbf{x}-\mathbf{G}\boldsymbol{\theta}}$$

- Thus the likelihood is:

$$L(\boldsymbol{\theta} \mid \mathbf{x}) = p(\mathbf{x} \mid \boldsymbol{\theta}) = p(\mathbf{x} - \mathbf{G}\boldsymbol{\theta}) \tag{8}$$

- Expanding this out we get

$$p(\mathbf{x} - \mathbf{G}\boldsymbol{\theta}) = \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left(-\frac{1}{2\sigma_e^2}(\mathbf{x} - \mathbf{G}\boldsymbol{\theta})^T(\mathbf{x} - \mathbf{G}\boldsymbol{\theta})\right)$$

Likelihood Estimation VII

- and taking logarithms:

$$\log L(\boldsymbol{\theta} \mid \mathbf{x}) = -(N/2)\log(2\pi\sigma_e^2) \; - \; \frac{1}{2\sigma_e^2}(\mathbf{x} - \mathbf{G}\boldsymbol{\theta})^T(\mathbf{x} - \mathbf{G}\boldsymbol{\theta})$$

- Thus maximisation of this function w.r.t. $\boldsymbol{\theta}$ is equivalent to *minimising* the quadratic term. This is exactly the criterion which is applied in the *ordinary least squares* (OLS) estimation method already considered.
- Hence we get immediately that the ML estimator is:

$$\boxed{\boldsymbol{\theta}^{\mathrm{ML}} = \boldsymbol{\theta}^{OLS} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\,\mathbf{x}} \tag{9}$$

Maximum likelihood for the Linear Gaussian model

Likelihood Estimation VIII

- In general, then, when the error process $\{e_n\}$ is zero-mean, independent and Gaussian with fixed variance, the OLS and ML solutions are *identical*.
- However, we would get a different solution if the noise were non-white and/or non-Gaussian - such models require a case-by-case ML analysis.
- Moreover, as we will see in a moment, the Bayesian inference method will give a new solution to the estimation problem, even in the white Gaussian noise case.

Estimating the variance I

- Finally, the noise variance can also be estimated in the Linear Gaussian Model by ML.
- To see this, look at the log-likelihood function at the optimal parameter estimate $\boldsymbol{\theta}^{ML}$ but now also considered as a function of σ_e^2:

$$\begin{aligned}\log L(\boldsymbol{\theta}^{ML}, \sigma_e^2 \mid \mathbf{x}) &= -(N/2)\log(2\pi\sigma_e^2) \\ &\quad - \frac{1}{2\sigma_e^2}(\mathbf{x} - \mathbf{G}\boldsymbol{\theta}^{ML})^T(\mathbf{x} - \mathbf{G}\boldsymbol{\theta}^{ML}) \\ &= -(N/2)\log(2\pi\sigma_e^2) - \frac{1}{2\sigma_e^2}J^{ML}\end{aligned}$$

where J^{ML} is the minimum squared error term corresponding to the ML optimisation.

Estimating the variance II

- Differentiate wrt σ_e^2 and set to zero to get:

$$\frac{\partial \log L(\boldsymbol{\theta}^{ML}, \sigma_e^2 \mid \mathbf{x})}{d\sigma_e^2} = -\frac{(N/2)}{\sigma_e^2} + \frac{J^{ML}}{2(\sigma_e^2)^2} = 0$$

and hence

$$\boxed{(\sigma_e^2)^{ML} = J^{ML}/N}$$

i.e. the just mean-squared error at the ML parameter solution.

Bayesian Methods I

- Thus far, in the Least squares and the ML methods, the parameter $\boldsymbol{\theta}$ of the GLM

$$\mathbf{x} = \mathbf{G}\boldsymbol{\theta} + \mathbf{e}$$

  was regarded as an unknown *constant.* The Bayesian approach treats $\boldsymbol{\theta}$ as a random vector.
- When treating $\boldsymbol{\theta}$ as a random vector, it must be assigned a joint p.d.f. for all its components. This p.d.f. is called the *prior* p.d.f.
- The p.d.f. for the parameter vector should ideally express the practitioner's knowledge about the relative probability of different parameter values *prior to, or before, the data is observed*.

Bayesian Methods II

- If nothing is known *a priori* about the parameters then the prior should be chosen to express no initial preference for one set of parameters over any other. (For example a uniform prior if appropriate for the model.)
- This willingness to assign priors which reflect subjective information is a powerful feature and also one of the most fundamental differences between the Bayesian and 'classical' (i.e. likelihood-based) inferential procedures.
- The precise form of probability density assigned *a priori* to the parameters requires careful consideration since misleading results can be obtained from an erroneous prior, but in principle at least we can apply the Bayesian approach to any problem where statistical uncertainty is present.

Bayesian Methods III

- Let

  $p(\boldsymbol{\theta})$ be the prior probability density of the parameters,
  $p(\mathbf{x} \mid \boldsymbol{\theta})$ is the p.d.f. of the data at parameter value $\boldsymbol{\theta}$.

- Bayes' Theorem gives the conditional p.d.f. of $\boldsymbol{\theta}$ given $\mathbf{x}$:

$$\boxed{p(\boldsymbol{\theta} \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})}{p(\mathbf{x})}} \tag{10}$$

- $p(\boldsymbol{\theta} \mid \mathbf{x})$ is also called the *posterior probability density function* of $\boldsymbol{\theta}$. Often abbreviated and just called the *posterior.*
- $p(\boldsymbol{\theta} \mid \mathbf{x})$ contains all the information the data $\mathbf{x}$ brings about the unobserved parameter $\boldsymbol{\theta}$.

Bayesian Methods IV

- The numerator in Bayes' theorem is the joint p.d.f. of $(\mathbf{x}, \boldsymbol{\theta})$ since

$$p(\mathbf{x}, \boldsymbol{\theta}) = p(\mathbf{x} \mid \boldsymbol{\theta}) \times p(\boldsymbol{\theta})$$

- The denominator is Bayes' theorem is the marginal p.d.f. of $\mathbf{x}$

$$p(\mathbf{x}) = \int p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) \, d\boldsymbol{\theta}$$

(in this and subsequent results the integration would be replaced by a summation in the case of a discrete valued random vector $\boldsymbol{\theta}$ which has a pmf and not a p.d.f. .)

- The denominator $p(\mathbf{x})$ in Statistical language is called the "evidence." It is an important quantity in the model selection problem, i.e. choosing between competing models for the data.

Bayesian Methods V

- Bayes' theorem is often stated in the form:

$$\boxed{p(\boldsymbol{\theta} \mid \mathbf{x}) \propto p(\mathbf{x} \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta})} \tag{11}$$

- The symbol $\propto$ means *proportional too.* What is missing in this version of Bayes' theorem is the normalising constant $p(\mathbf{x})$ in the denominator which makes $p(\boldsymbol{\theta} \mid \mathbf{x})$ a p.d.f. in $\boldsymbol{\theta}$, that is $\int p(\boldsymbol{\theta} \mid \mathbf{x}) \mathrm{d}\boldsymbol{\theta} = 1$.

Bayesian Methods VI

- Before $\mathbf{x}$ is observed $p(\boldsymbol{\theta})$ expresses any information previously obtained concerning $\boldsymbol{\theta}$.
- Any new information concerning the parameters contained in $\mathbf{x}$ is then given by $p(\boldsymbol{\theta} \mid \mathbf{x})$.
- Intuitively, the application of Bayes' theorem can be viewed as a transformation of the prior to the posterior. This transformation can be thought of as a refinement of any previous ('prior') knowledge about the parameters once having observed the data $\mathbf{x}$.
- Clearly if we start off with little or no information about $\boldsymbol{\theta}$ then the posterior is likely to obtain information almost solely from $\mathbf{x}$. (Think of the case when $p(\boldsymbol{\theta})$ is the uniform p.d.f.)

Bayesian Methods VII

- Conversely, if $p(\boldsymbol{\theta})$ already expresses a significant amount of information about $\boldsymbol{\theta}$ then $\mathbf{x}$ will contribute relatively less new information to the posterior. (Think of the case when the prior is a "narrow" Gaussian density.)
- It is these attributes that make $p(\boldsymbol{\theta} \mid \mathbf{x})$ a more rigorous approach for parameter estimation compared to a Likelihood based approach. (Recall that the likelihood $L(\boldsymbol{\theta} \mid \mathbf{x})$ expresses how "likely" each θ value is for the data $\mathbf{x}$ and is not a probability density for $\boldsymbol{\theta}$.)
- In both the ML and Bayesian approach, if the p.d.f. we assumed for the data generation is incorrect, we cannot provide any guarantees for our answers and they may indeed be skewed but the incorrect p.d.f. assumed. In contrast, the Least squares approach makes no assumptions on the probability model thus is potentially more robust to modelling

Bayesian Methods VIII

errors. Unfortunately, like the Likelihood approach, least squares does not incorporate prior knowledge about the parameters.

- We may in principle manipulate the posterior density to infer any required statistic of $\boldsymbol{\theta}$ conditional upon $\mathbf{x}$.
- This is a significant advantage over ML and least squares methods which strictly give us only a single estimate of $\boldsymbol{\theta}$, known as a 'point estimate'.
- However, by producing a posterior p.d.f. with values defined for all $\boldsymbol{\theta}$ the Bayesian approach gives a fully interpretable probability distribution.
- In principle this is as much as one could ever need to know about the inference problem.

Computing the posterior p.d.f. I

We now work through computing $p(\mathbf{x} \mid \boldsymbol{\theta})$ when $\mathbf{x}$ is a linear function of $\boldsymbol{\theta}$ with additive Gaussian noise that is

$$\mathbf{x} = \mathbf{G}\boldsymbol{\theta} + \mathbf{e}, \qquad \text{where } \mathbf{e} \text{ is a Gaussian noise vector}$$

and the prior for $\boldsymbol{\theta}$ is also Gaussian.

- Let

$$\begin{aligned} p(\boldsymbol{\theta}) &= \mathcal{N}(\mathbf{m}, \mathbf{C}) \\ &= \frac{1}{(2\pi)^{P/2} \mid \mathbf{C} \mid^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \mathbf{m})^T \mathbf{C}^{-1} (\boldsymbol{\theta} - \mathbf{m})\right) \end{aligned}$$

where $\mathbf{m}$ is the prior parameter mean vector, $\mathbf{C}$ is the parameter covariance matrix and P is the number of parameters in $\boldsymbol{\theta}$.

Computing the posterior p.d.f. II

- The likelihood $p(\mathbf{x} \mid \boldsymbol{\theta})$ takes the same form as before for the ML estimator, so the posterior is as follows:

$$p(\boldsymbol{\theta} \mid \mathbf{x}) \propto p(\boldsymbol{\theta}) p(\mathbf{x} \mid \boldsymbol{\theta})$$

$$= \frac{1}{(2\pi)^{P/2} \mid \mathbf{C} \mid^{1/2}} \exp\left(-\frac{1}{2} (\boldsymbol{\theta} - \mathbf{m})^T \mathbf{C}^{-1} (\boldsymbol{\theta} - \mathbf{m}) \right)$$

Prior

Likelihood

$$\frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left(-\frac{1}{2\sigma_e^2} (\mathbf{x} - \mathbf{G}\boldsymbol{\theta})^T (\mathbf{x} - \mathbf{G}\boldsymbol{\theta}) \right) \quad (12)$$

Computing the posterior p.d.f. III

- So, -2 times the log-density is given by:

$$-2 \times \log(p(\boldsymbol{\theta} \mid \mathbf{x})) = (\boldsymbol{\theta} - \mathbf{m})^T \mathbf{C}^{-1} (\boldsymbol{\theta} - \mathbf{m}) + \frac{1}{\sigma_e^2} (\mathbf{x} - \mathbf{G}\boldsymbol{\theta})^T (\mathbf{x} - \mathbf{G}\boldsymbol{\theta}) + \text{terms without } \boldsymbol{\theta}$$

- If we can further simplify this to

$$-2 \times \log(p(\boldsymbol{\theta} \mid \mathbf{x})) = (\boldsymbol{\theta} - \bar{\mathbf{m}})^T \bar{\mathbf{C}}^{-1} (\boldsymbol{\theta} - \mathbf{m}) + \text{terms without } \boldsymbol{\theta}$$

for some vector $\bar{\mathbf{m}}$ and a symmetric positive definite matrix $\bar{\mathbf{C}}$ then $p(\boldsymbol{\theta} \mid \mathbf{x})$ is a Gaussian density.

Computing the posterior p.d.f. IV

- $\bar{\mathbf{m}}$ is clearly the maximiser of $\log(p(\boldsymbol{\theta} \mid \mathbf{x}))$

$$\boxed{\boldsymbol{\theta}^{\text{MAP}} = \arg\max_{\theta} \log p(\boldsymbol{\theta} \mid \mathbf{x})} \tag{13}$$

Maximum a posteriori (MAP) estimator

- The maximiser $\boldsymbol{\theta}^{\text{MAP}}$ is obtained by differentiation and setting to 0 in a similar way as for the ML approach:

Computing the posterior p.d.f. V

$$\boxed{\boldsymbol{\theta}^{\text{MAP}} = \left(\mathbf{G}^T\mathbf{G} + \sigma_e^2\,\mathbf{C}^{-1}\right)^{-1}\left(\mathbf{G}^T\mathbf{x} + \sigma_e^2\,\mathbf{C}^{-1}\mathbf{m}\right)} \qquad (14)$$

MAP estimator - Linear Gaussian model

- Rearrange the exponent of (12) by 'completing the square', as we did for the likelihood function earlier:

Computing the posterior p.d.f. VI

$$\begin{aligned}
&\frac{1}{\sigma_e^2}(\mathbf{x}-\mathbf{G}\boldsymbol{\theta})^T(\mathbf{x}-\mathbf{G}\boldsymbol{\theta})+(\boldsymbol{\theta}-\mathbf{m})^T\mathbf{C}^{-1}(\boldsymbol{\theta}-\mathbf{m}) \\
&\quad=\frac{1}{\sigma_e^2}\left((\boldsymbol{\theta}-\boldsymbol{\theta}^{\mathrm{MAP}})^T\boldsymbol{\Phi}(\boldsymbol{\theta}-\boldsymbol{\theta}^{\mathrm{MAP}})\right) \\
&\qquad+\frac{1}{\sigma_e^2}\left(\mathbf{x}^T\mathbf{x}+\sigma_e^2\mathbf{m}^T\mathbf{C}^{-1}\mathbf{m}-\mathbf{h}^T\boldsymbol{\theta}^{\mathrm{MAP}}\right) \qquad (15)
\end{aligned}$$

with terms defined as

$$\begin{aligned}
\boldsymbol{\theta}^{\mathrm{MAP}} &= \boldsymbol{\Phi}^{-1}\mathbf{h} \\
\boldsymbol{\Phi} &= \mathbf{G}^T\mathbf{G}+\sigma_e^2\mathbf{C}^{-1} \\
\mathbf{h} &= \mathbf{G}^T\mathbf{x}+\sigma_e^2\mathbf{C}^{-1}\mathbf{m}
\end{aligned}$$

Computing the posterior p.d.f. VII

- Now we can observe that the first term in (15),

$$\frac{1}{\sigma_e^2}\left((\boldsymbol{\theta}-\boldsymbol{\theta}^{\text{MAP}})^T\boldsymbol{\Phi}(\boldsymbol{\theta}-\boldsymbol{\theta}^{\text{MAP}})\right),$$

  is in exactly the correct form for the exponent of a multivariate Gaussian, with mean vector and covariance matrix as follows,

$$\text{mean} = \boldsymbol{\theta}^{\text{MAP}}, \qquad \text{covariance matrix} = \sigma_e^2\boldsymbol{\Phi}^{-1}.$$

- Since the remaining terms in (15) do not depend on $\boldsymbol{\theta}$, and we know that the multivariate density function must be proper (i.e. integrate to 1), we can conclude that the posterior distribution is itself a multivariate Gaussian,

$$p(\boldsymbol{\theta} \mid \mathbf{x}) = \mathcal{N}(\boldsymbol{\theta}^{\text{MAP}}, \sigma_e^2\boldsymbol{\Phi}^{-1}). \tag{16}$$

  This formula will allow us to reinterpret the Bayesian estimator for the linear model in terms of its mean-squared error.

Example: Gaussian Model with one observation I

When $P = 1$ and $N = 1$, take $\mathbf{G} = 1$ so that:

$$x = \theta + e$$

and prior:

$$p(\theta) = \mathcal{N}(\mu_\theta, \sigma_\theta^2)$$

The likelihood is just:

$$p(x|\theta) = \mathcal{N}(x|\theta, \sigma_e^2) \propto e^{-\frac{1}{2\sigma_e^2}(x-\theta)^2} = e^{-\frac{1}{2\sigma_e^2}(\theta-x)^2}$$

Note that this is Gaussian-shaped as a function of either x or θ. Using the above formulae, or re-deriving by hand, we get:

$$\theta^{MAP} = \frac{x + \sigma_e^2 \mu_\theta / \sigma_\theta^2}{1 + \sigma_e^2 / \sigma_\theta^2} = \frac{\sigma_\theta^2 x + \sigma_e^2 \mu_\theta}{\sigma_\theta^2 + \sigma_e^2}$$

Example: Gaussian Model with one observation II

and

$$\text{variance} = \frac{\sigma_e^2}{1 + \sigma_e^2/\sigma_\theta^2} = \frac{\sigma_e^2 \sigma_\theta^2}{\sigma_\theta^2 + \sigma_e^2}$$

So,

$$p(\theta|x) = \mathcal{N}\left(\frac{\sigma_\theta^2 x + \sigma_e^2 \mu_\theta}{\sigma_\theta^2 + \sigma_e^2}, \frac{\sigma_e^2 \sigma_\theta^2}{\sigma_\theta^2 + \sigma_e^2}\right)$$

See plots for different prior-likelihood trade-offs...

Example: Gaussian Model with one observation III

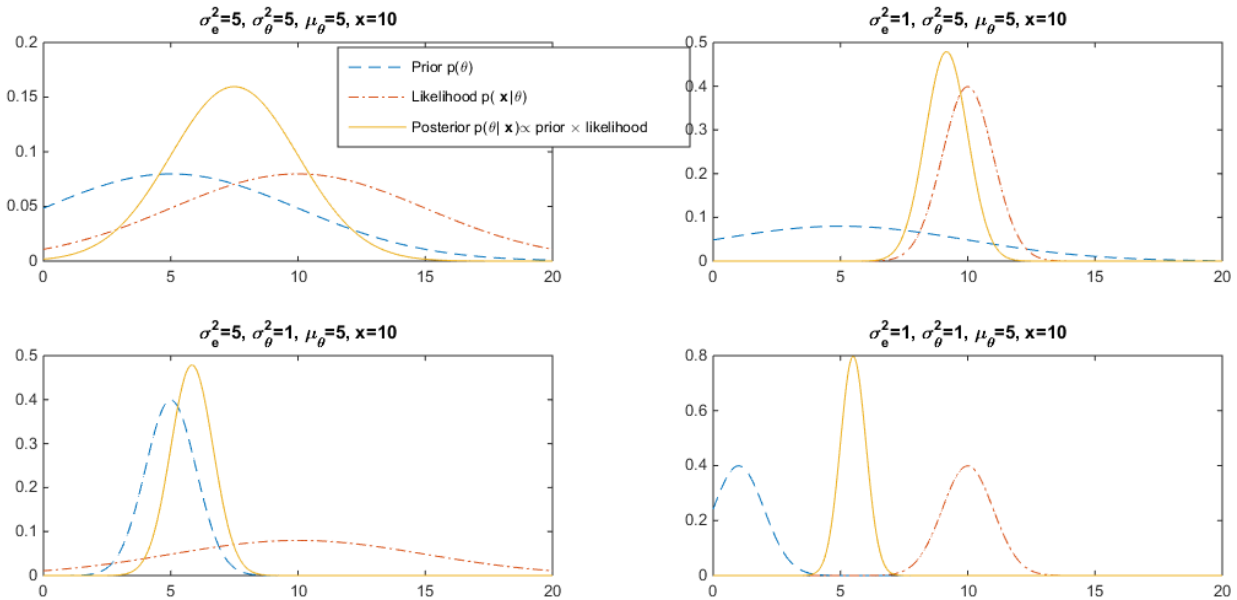


Figure 4 : Prior/ likelihood combinations - Linear Gaussian model, P=1, N=1. Top left: Prior/likelihood balanced; Top Right: Likelihood dominates; Bottom left: Prior dominates; Bottom right: prior-likelihood conflict

Further comments I

- Compare MAP estimate in (14) directly with the ML estimator:

$$\boldsymbol{\theta}^{\mathrm{ML}} = \boldsymbol{\theta}^{OLS} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\,\mathbf{x} \tag{17}$$

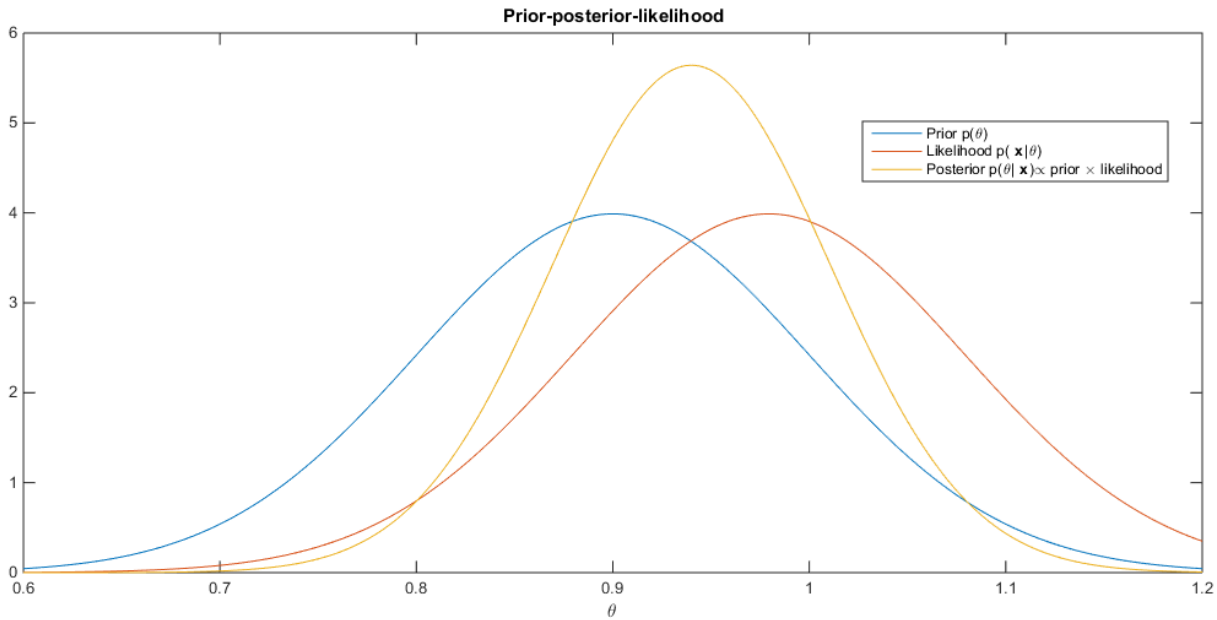


Figure 5 : Prior, likelihood and posterior for 1-D Gaussians

Further comments II

- In these expressions we can clearly see the 'regularising' effect of the prior density on the ML estimate of (9). As the prior becomes more 'diffuse', i.e. the diagonal elements of $\mathbf{C}$ increase both in magnitude and relative to the off-diagonal elements, we impose 'less' prior information on the estimate. In the limit the prior tends to a uniform ('flat') prior with all $\boldsymbol{\theta}$ equally probable. In this limit $\mathbf{C}^{-1} = 0$ and the estimate is identical to the ML estimate (9). This useful relationship demonstrates that the ML estimate may be interpreted as the MAP estimate with a uniform prior assigned to $\boldsymbol{\theta}$.
- The MAP estimate will also tend towards the ML estimate when the likelihood is strongly 'peaked' around its maximum compared with the prior. Once again the prior will then have little influence on the shape of the posterior density. It is in fact well known that as the sample size N tends to infinity the Bayes solution tends to the ML solution.

Further comments III

- This of course says nothing about small sample (i.e. small N) parameter estimates where the effect of the prior may be very significant.
- The choice of a multivariate Gaussian prior may well be motivated by physical considerations about the problem, or it may be motivated by subjective prior knowledge about the value of $\boldsymbol{\theta}$ (before the data $\mathbf{x}$ are seen!) in terms of a rough value $\mathbf{m}$ and a confidence in that value through the covariance matrix $\mathbf{C}$ (a 'subjective' prior). In fact the choice of Gaussian also has the very special property that it makes the Bayesian calculations straightforward and available in closed form. Such a prior is known as a 'conjugate' prior.

MMSE Estimation I

- Thus far we have derived the posterior $p(\boldsymbol{\theta} \mid \mathbf{x})$ for the specific data vector $\mathbf{x}$ observed.
- We are often also required to give a single point estimate for $\boldsymbol{\theta}$ from $p(\boldsymbol{\theta} \mid \mathbf{x})$.
- Here are two examples of point estimates extracted from $p(\boldsymbol{\theta} \mid \mathbf{x})$:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\theta} p(\boldsymbol{\theta} \mid \mathbf{x})$$

$$\text{or} \qquad \hat{\boldsymbol{\theta}} = \int \boldsymbol{\theta} p(\boldsymbol{\theta} \mid \mathbf{x}) \mathrm{d}\boldsymbol{\theta}$$

- Given that we "extracted" $\hat{\boldsymbol{\theta}}$ from $p(\boldsymbol{\theta} \mid \mathbf{x})$, in both cases $\hat{\boldsymbol{\theta}}$ is a function of the observed data $\mathbf{x}$.
- The first point estimate is the MAP estimate and the second is the mean value of the p.d.f. $p(\boldsymbol{\theta} \mid \mathbf{x})$.

MMSE Estimation II

- The optimality of these estimates can be justified. A principled way of choosing the best point estimate is by introducing a cost function $C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})$. Some books say *loss function* instead.
- $C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})$ expresses the cost of estimating the parameter as $\hat{\boldsymbol{\theta}}$ when the true value is $\boldsymbol{\theta}$.
- A cost function should be non-negative and satisfy $C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) = 0$.
- We then choose $\hat{\boldsymbol{\theta}}$ as the minimiser of

$$J(\boldsymbol{\theta}') = \int C(\boldsymbol{\theta}', \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathbf{x}) \mathrm{d}\boldsymbol{\theta}$$

- The choice of cost function will depend on the requirements of a particular problem.

MMSE Estimation III

- Assume $\boldsymbol{\theta}$ is scalar valued (we thus write θ instead) and examples of cost are

$$\begin{aligned}
\text{Squared error}: &\quad C(\theta', \theta) = (\theta' - \theta)^2, \\
\text{Absolute error}: &\quad C(\theta', \theta) = |\theta' - \theta|, \\
\text{Zero} - \text{one error}: &\quad C(\theta', \theta) = \begin{cases} 0, & \text{if } \theta' = \theta, \\ 1 & \text{otherwise.} \end{cases}
\end{aligned}$$

- We focus on the square error and its best estimate is

$$\hat{\theta}^{\text{MMSE}} = \arg\min_{\theta'} \int (\theta' - \theta)^2 p(\theta|\mathbf{x}) d\theta$$

The resulting estimate is called Minimum mean-squared error (MMSE) estimate.

MMSE Estimation IV

- The cost can be expressed more tersely as

$$\mathbb{E}[(\theta' - \theta)^2 \mid \mathbf{x}] = \int (\theta' - \theta)^2 p(\theta|\mathbf{x}) d\theta$$

- Differentiating with respect to θ' gives

$$\frac{d}{d\theta'} \int (\theta' - \theta)^2 p(\theta|\mathbf{x}) d\theta = \int \frac{d}{d\theta'} (\theta' - \theta)^2 p(\theta|\mathbf{x}) d\theta$$
$$= \int 2(\theta' - \theta) p(\theta|\mathbf{x}) d\theta$$

and setting to zero we get the minimiser:

$$\int_\theta 2(\theta' - \theta) p(\theta|\mathbf{x}) d\theta = 0$$

or,

$$\theta' = \int \theta p(\theta|\mathbf{x}) d\theta = \mathbb{E}[\theta \mid \mathbf{x}]$$

MMSE Estimation V

- Recap, the MMSE estimate is the best estimate for the squared error:

$$\boxed{\hat{\theta}^{\text{MMSE}} = \mathbb{E}[\theta|\mathbf{x}] = \int \theta p(\theta|\mathbf{x}) d\theta}$$

MMSE for the linear Gaussian model I

The posterior distribution was obtained as:

$$p(\boldsymbol{\theta} \mid \mathbf{x}) = \mathcal{N}(\boldsymbol{\theta}^{\text{MAP}}, \sigma_e^2 \boldsymbol{\Phi}^{-1}). \tag{18}$$

The mean value of this distribution is of course $\boldsymbol{\theta}^{\text{MAP}}$.
Hence the MMSE estimator for the Linear Gaussian model is, conveniently:

$$\boldsymbol{\theta}^{MMSE} = \boldsymbol{\theta}^{\text{MAP}}$$

In other cases, the estimators do not necessarily coincide - see Examples paper.

Summary of Estimators I

- We have examined three important estimation methods, each of increasing sophistication: Ordinary Least Squares (OLS), Maximum Likelihood (ML) and the estimate derived through the use of a loss or cost function (e.g. the MMSE estimate.)
- The OLS was shown to have some optimality: it was the best linear unbiased estimate. (Recall the assumptions in the derivation?)
- The ML estimate was derived but we did not state any optimality. The ML estimate will in fact converge to the true parameter in the limit of large amounts of data whereas the OLS will not in general.
- The MMSE estimate is optimal for the squared error loss function. Changing the loss function will change the best estimate to something else, e.g. the MAP estimate is best when the loss function is the zero-one loss function.

Summary of Estimators II

- It was seen that the OLS estimate coincides with the ML estimate when the noise vector is comprised of independent and identically distributed zero-mean Gaussian random variables.
- In turn the ML estimate coincided with the Maximum *a posteriori* (MAP) estimator when the prior distribution on θ is uniform.
- Unlike the OLS approach, both the ML and MAP requires specific knowledge of the likelihood function and the MAP estimate, being extracted from the posterior, requires knowledge of the prior density $p(\theta)$ as well.
- The choice of which estimate to use will thus depend on the degree of knowledge available and the performance required.

Summary of Estimators III

We can now summarize the performance, advantages (+) and disadvantages (-) of the three schemes:

- **Least Squares (LS)**
  - Requires no knowledge of probability distributions (+)
  - Cannot incorporate prior knowledge about parameter probability distributions (-)
  - Usually the simplest scheme to implement (+)
  - Guarantee of performance as BLUE estimator (+)
  - No guarantees of performance compared to nonlinear estimators
- **Maximum Likelihood (ML)**
  - Requires knowledge of noise (model) probability distribution (-)
  - Cannot incorporate prior knowledge about parameter probability distributions (-)
  - Can be more complicated to implement than LS in the non-Gaussian case (-)

Summary of Estimators IV

- Performance guaranteed to be optimal when the amount of data is large (+)

- **Bayesian (MAP and MMSE)**
  - Requires knowledge of noise (model) probability distribution (-)
  - Requires knowledge of parameter prior probability distribution (might require subjective input) (-)
  - Incorporates prior knowledge about parameter probability distribution (+)
  - Can be more complicated to implement than LS or ML, depending on form of likelihood and prior (-)
  - Performance guaranteed to be optimal for any amount of data (*provided* prior distribution is correct) (+/-)