Parameter Estimation

Lecture 15

Draft

- Least Squares
- Exercises
- Recursive Least Squares
- Recursive Least Squares with Forgetting Factor

Least Squares
Lecture 15

Draft

A data model with m inputs and p parameters, one output, a set of n measurements (with the corresponding inputs) and the residual function to minimize:

- $f : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$
$$y = f(x_1, x_2, \ldots, x_m, \beta_1, \beta_2, \ldots, \beta_p)$$
- $\mathcal{Y} \subseteq \mathbb{R}^m \times \mathbb{R}$
$$\mathcal{Y} = \{(x_{1,1}, \ldots, x_{m,1}, \tilde{y}_1), \ldots, (x_{1,n}, \ldots, x_{m,n}, \tilde{y}_n)\}$$
- $J(\beta_1, \ldots, \beta_p) = \sum_{k=1}^{n} \left(f(x_{1,k}, \ldots, x_{m,k}, \beta_1, \ldots, \beta_p) - \tilde{y}_k \right)^2$
$$(\beta_1^*, \ldots, \beta_p^*) = \arg\min J$$

Least Squares
Lecture 15

Draft

Since the model has p parameters, there are p gradient equations:

$$\frac{\partial J}{\partial \beta_j} = 2\sum_{k}\Big(f(x_{1,k},\ldots,x_{n,k},\beta_1,\ldots,\beta_p) - \tilde{y}_k\Big)\frac{\partial f}{\partial \beta_j} = 0; \quad j = \overline{1,p}$$

The gradient equations apply to all least squares problems. Each particular problem requires particular expressions for the model function f and thus its partial derivatives.

Linear Least Squares
Lecture 15

Draft

- The data model is linear in the parameters:
  $y = f_1(x_1,\dots,x_m)\cdot\beta_1 + \dots + f_p(x_1,\dots,x_m)\cdot\beta_p, f_i : \mathbb{R}^m \to \mathbb{R}, i \in \overline{1,p}$
  $y = f_1(\mathbf{X})\cdot\beta_1 + \dots + f_p(\mathbf{X})\cdot\beta_p = \sum_{i=1}^{p} f_i(\mathbf{X})\cdot\beta_i$
- The residual function J:
  $J(\beta_1,\dots,\beta_p) = \sum_{k=1}^{n}\left(\tilde{y}_k - \sum_{i=1}^{p} f_i(\mathbf{X}_k)\cdot\beta_i\right)^2$
- The minimization of the residual function J is equivalent to:

$$\frac{\partial J}{\partial \beta_j} = -2\sum_{k=1}^{n}\left(\tilde{y}_k - \sum_{i=1}^{p} f_i(\mathbf{X}_k)\cdot\beta_i\right) f_j(\mathbf{X}_k) = 0$$

Linear Least Squares

Lecture 15

Draft

Express the residual function into a matrix form:

$$J(\beta_1, \ldots, \beta_p) = \sum_{k=1}^{n} \left(\tilde{y}_k - \sum_{i=1}^{p} f_i(\mathbf{X}_k) \cdot \beta_i \right)^2$$

We will use:

- $\sum_{k=1}^{p} a_k^2 = \begin{bmatrix} a_1 & \ldots & a_p \end{bmatrix} \begin{bmatrix} a_1 \\ \ldots \\ a_p \end{bmatrix} = \mathbf{a}^T \mathbf{a}$
- $\tilde{y}_k - \sum_{i=1}^{p} f_i(\mathbf{X}_k) \cdot \beta_i = \tilde{y}_k - \begin{bmatrix} f_1(\mathbf{X}_k) & \ldots & f_p(\mathbf{X}_k) \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}$

Linear Least Squares

Lecture 15

Draft

Express the residual function into a matrix form:

$$J(\beta_1, \ldots, \beta_p) = (\mathbf{Y} - \mathbf{F}_x \boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{F}_x \boldsymbol{\beta}),$$

where $\mathbf{Y} = \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_k \end{bmatrix}$, $\mathbf{F}_x = \begin{bmatrix} f_1(\mathbf{X}_1) & \ldots & f_p(\mathbf{X}_1) \\ f_1(\mathbf{X}_2) & \ldots & f_p(\mathbf{X}_2) \\ \vdots & & \\ f_1(\mathbf{X}_n) & \ldots & f_p(\mathbf{X}_n) \end{bmatrix}$ and $\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}$

Linear Least Squares

Lecture 15

So let's find the solution of the linear least square:

$$\frac{\partial J(\beta)}{\partial \beta} = \frac{\partial \left((\mathbf{Y} - \mathbf{F}_x \beta)^T (\mathbf{Y} - \mathbf{F}_x \beta) \right)}{\partial \beta} = 0$$

$$\frac{\partial \left(\mathbf{Y}^T \mathbf{Y} - \mathbf{Y}^T \mathbf{F}_x \beta - \beta^T \mathbf{F}_x^T \mathbf{Y} + \beta^T \mathbf{F}_x^T \mathbf{F}_x \beta \right)}{\partial \beta} = 0$$

$$-\mathbf{Y}^T \mathbf{F}_x - \mathbf{Y}^T \mathbf{F}_x + 2\beta^T \mathbf{F}_x^T \mathbf{F}_x = 0$$

$$\boxed{\beta = \left(\mathbf{F}_x^T \mathbf{F}_x \right)^{-1} \mathbf{F}_x^T \mathbf{Y}}$$

Draft

Matrix/Multivariate Calculus

Lecture 15

Draft

https://en.wikipedia.org/wiki/Matrix_calculus

Identities: vector-by-vector $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$

Condition	Expression	Numerator layout, i.e. by $\mathbf{y}$ and $\mathbf{x}^\top$	Denominator layout, i.e. by $\mathbf{y}^\top$ and $\mathbf{x}$
$\mathbf{a}$ is not a function of $\mathbf{x}$	$\frac{\partial \mathbf{a}}{\partial \mathbf{x}} =$	$\mathbf{0}$	
	$\frac{\partial \mathbf{x}}{\partial \mathbf{x}} =$	$\mathbf{I}$	
$\mathbf{A}$ is not a function of $\mathbf{x}$	$\frac{\partial \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} =$	$\mathbf{A}$	$\mathbf{A}^\top$
$\mathbf{A}$ is not a function of $\mathbf{x}$	$\frac{\partial \mathbf{x}^\top \mathbf{A}}{\partial \mathbf{x}} =$	$\mathbf{A}^\top$	$\mathbf{A}$
a is not a function of $\mathbf{x}$, $\mathbf{u} = \mathbf{u}(\mathbf{x})$	$\frac{\partial a\mathbf{u}}{\partial \mathbf{x}} =$	$a\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	
$v = v(\mathbf{x})$, $\mathbf{a}$ is not a function of $\mathbf{x}$	$\frac{\partial v\mathbf{a}}{\partial \mathbf{x}} =$	$\mathbf{a}\frac{\partial v}{\partial \mathbf{x}}$	$\frac{\partial v}{\partial \mathbf{x}}\mathbf{a}^\top$
$v = v(\mathbf{x})$, $\mathbf{u} = \mathbf{u}(\mathbf{x})$	$\frac{\partial v\mathbf{u}}{\partial \mathbf{x}} =$	$v\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \mathbf{u}\frac{\partial v}{\partial \mathbf{x}}$	$v\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial v}{\partial \mathbf{x}}\mathbf{u}^\top$
$\mathbf{A}$ is not a function of $\mathbf{x}$, $\mathbf{u} = \mathbf{u}(\mathbf{x})$	$\frac{\partial \mathbf{A}\mathbf{u}}{\partial \mathbf{x}} =$	$\mathbf{A}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\mathbf{A}^\top$
$\mathbf{u} = \mathbf{u}(\mathbf{x})$, $\mathbf{v} = \mathbf{v}(\mathbf{x})$	$\frac{\partial (\mathbf{u}+\mathbf{v})}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$	
$\mathbf{u} = \mathbf{u}(\mathbf{x})$	$\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}$
$\mathbf{u} = \mathbf{u}(\mathbf{x})$	$\frac{\partial \mathbf{f}(\mathbf{g}(\mathbf{u}))}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{f}(\mathbf{g})}{\partial \mathbf{g}}\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{f}(\mathbf{g})}{\partial \mathbf{g}}$

Excercise 1

Lecture 15

Draft

Linear Least Squares on 2D data:
Given the times of the last six men and women Olympic winners of the 100 meters sprint, predict the time of the next champion at Paris 2024.

Year	S2000	A2004	B2008	L2012	R2016	T2020
Women	11.12	10.93	10.78	10.75	10.71	10.61
Men	9.87	09.85	9.69	9.63	9.81	9.80

Data from olympics.com

Exercise 1

Lecture 15

- Choose inputs, choose the data model, e.g. $f(x_1, x_2) = ((x_1 - 1924)/4)^w \cdot \beta_1 + x_2 \cdot \beta_2,\ x_2 = 1,\ w \in [-1, 1]$
- There are $m = 6$ measurements, $n = 2$ parameters
- $\mathbf{Y} = \begin{bmatrix} 11.12 \\ 10.93 \\ 10.78 \\ 10.75 \\ 10.71 \\ 10.61 \end{bmatrix}, \mathbf{F}_x = \begin{bmatrix} 19 & 1 \\ 20 & 1 \\ 21 & 1 \\ 22 & 1 \\ 23 & 1 \\ 24 & 1 \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}$
- $\boldsymbol{\beta} = \left(\mathbf{F}_x^T \mathbf{F}_x\right)^{-1} \mathbf{F}_x^T \mathbf{Y}$

Draft

Exercise 2
Lecture 15

Draft

- System Identification Example, ARX model:

$$y_k = \sum_{i=1}^{p_a} \alpha_i y_{k-i} + \sum_{j=1}^{p_b} \beta_j u_{k-j-1}$$

- ARX-like models nonlinear in state and inputs (but linear in parameters), e.g:

$$y_k = \beta_1 y_{k-1} y_{k-2} + \beta_2 u_k \sin(y_{k-1}) + \dots$$

- Total Least-Squares, when there are measurement errors in the "inputs" (matrix $\mathbf{F}_x$) - Discussion

Quality of the Estimation
Lecture 15

Draft

- Do the residuals look like random noise?
- Coefficient of Determination, i.e. R^2:

$$R^2 = 1 - \frac{\sum_{k=1}^{n}\left(\tilde{y}_k - f(x_{1,k},\ldots,x_{m,k},\beta_1,\ldots,\beta_n)\right)^2}{\sum_{k=1}^{n}\left(\tilde{y}_k - \bar{y}\right)^2}$$

- Evaluate the quality of the estimation for the two examples

Recursive Least Squares
Lecture 15

Draft

- Let us assume we have just calculated the solution of a LS problem, from a given dataset:

$$\beta = \left(\mathbf{F}_x^T \mathbf{F}_x\right)^{-1} \mathbf{F}_x^T \mathbf{Y},$$

where $\mathbf{Y} = \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_n \end{bmatrix}$, $\mathbf{F}_x = \begin{bmatrix} f_1(\mathbf{X}_1) & \dots & f_p(\mathbf{X}_1) \\ f_1(\mathbf{X}_2) & \dots & f_p(\mathbf{X}_2) \\ \vdots & & \\ f_1(\mathbf{X}_n) & \dots & f_p(\mathbf{X}_n) \end{bmatrix}$ and $\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}$

- And we just received another measurement point:

$$\tilde{y}_{n+1}, f_1(\mathbf{X}_{n+1}), \dots, f_p(\mathbf{X}_{n+1}),$$

Recursive Least Squares
Lecture 15

Draft

- So what is the solution?

$$\hat{\beta} = \left(\mathbf{F}_x^T(n+1)\mathbf{F}_x(n+1)\right)^{-1}\mathbf{F}_x^T(n+1)\mathbf{Y}(n+1),$$

where $\mathbf{Y} = \begin{bmatrix} \mathbf{Y}(n) \\ \tilde{y}_{n+1} \end{bmatrix}$, $\mathbf{F}_x = \begin{bmatrix} \mathbf{F}_x(n) & & \\ f_1(\mathbf{X}_{n+1}) & \dots & f_p(\mathbf{X}_{n+1}) \end{bmatrix} = \begin{bmatrix} \mathbf{F}_x(n) \\ \mathbf{f}^T(n+1) \end{bmatrix}$

- But let's work on this a bit, because we do not want to do the inversion again from scratch

Recursive Least Squares

Lecture 15

Draft

$$\beta = \left(\begin{bmatrix} \mathbf{F}_x^T(n) & \mathbf{f}(n+1) \end{bmatrix} \begin{bmatrix} \mathbf{F}_x(n) \\ \mathbf{f}^T(n+1) \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{F}_x^T(n) & \mathbf{f}(n+1) \end{bmatrix} \begin{bmatrix} \mathbf{Y}(n) \\ \tilde{y}_{n+1} \end{bmatrix} =$$

$$= \left(\mathbf{F}_x^T(n)\mathbf{F}_x(n) + \mathbf{f}(n+1)\mathbf{f}^T(n+1) \right)^{-1} \begin{bmatrix} \mathbf{F}_x^T(n) & \mathbf{f}(n+1) \end{bmatrix} \begin{bmatrix} \mathbf{Y}(n) \\ \tilde{y}_{n+1} \end{bmatrix}$$

- (Woodbury) Matrix Inversion Lemma:

$$(\mathbf{A} + \mathbf{UCV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1},$$

where $\mathbf{A}$ is $n \times n$, $\mathbf{C}$ is $k \times k$, $\mathbf{U}$ is $n \times k$ and $\mathbf{V}$ is $k \times n$ and $\mathbf{A}$ and $\mathbf{C}$ are invertible.

Recursive Least Squares

Lecture 15

Draft

$$\mathbf{P}_x(n+1) \triangleq \left(\mathbf{F}_x^T(n+1)\mathbf{F}_x(n+1)\right)^{-1} = \left(\mathbf{F}_x^T(n)\mathbf{F}_x(n) + \mathbf{f}(n+1)\mathbf{f}^T(n+1)\right)^{-1} =$$

$$= \mathbf{P}_x(n) - \mathbf{P}_x(n)\mathbf{f}(n+1)\left(1 + \mathbf{f}^T(n+1)\mathbf{P}_x(n)\mathbf{f}(n+1)\right)^{-1}\mathbf{f}^T(n+1)\mathbf{P}_x(n) =$$

$$= \mathbf{P}_x(n) - \frac{\mathbf{P}_x(n)\mathbf{f}(n+1)\mathbf{f}^T(n+1)\mathbf{P}_x(n)}{1 + \mathbf{f}^T(n+1)\mathbf{P}_x(n)\mathbf{f}(n+1)}$$

where $\mathbf{P}_x(n)$ is of course $\left(\mathbf{F}_x^T(n)\mathbf{F}_x(n)\right)^{-1}$.

Recursive Least Squares
Lecture 15

Draft

- So,

$$\beta(n) = \mathbf{P}_x(n)\mathbf{F}_x^T(n)\tilde{\mathbf{y}}(n); \; \mathbf{P}_x(n) = \left(\mathbf{F}_x^T(n)\mathbf{F}_x(n)\right)^{-1}$$

$$\beta(n+1) = \mathbf{P}_x(n+1)\mathbf{F}_x^T(n+1)\tilde{\mathbf{y}}(n+1), \text{ where}$$

$$\mathbf{P}_x(n+1) = \mathbf{P}_x(n) - \frac{\mathbf{P}_x(n)\mathbf{f}(n+1)\mathbf{f}^T(n+1)\mathbf{P}_x(n)}{1 + \mathbf{f}^T(n+1)\mathbf{P}_x(n)\mathbf{f}(n+1)}$$

- More over, we can write $\beta(n+1)$ as:

$$\beta(n+1) = \beta(n) + \frac{\mathbf{P}_x(n)\mathbf{f}(n+1)}{1 + \mathbf{f}^T(n+1)\mathbf{P}_x(n)\mathbf{f}(n+1)}\left(\tilde{y}(n+1) - \mathbf{f}^T(n+1)\beta(n)\right)$$

Recursive Least Squares

Lecture 15

This form is important. Let call $\mathbf{K} = \frac{\mathbf{P}_x(n)\mathbf{f}(n+1)}{1+\mathbf{f}^T(n+1)\mathbf{P}_x(n)\mathbf{f}(n+1)}$ and $\hat{y}(n+1) = \mathbf{f}^T(n+1)\beta(n)$. Then

$$\beta(n+1) = \beta(n) + \mathbf{K}\Big(\tilde{y}(n+1) - \hat{y}(n+1)\Big)$$

Draft

- The equation can be though of as a correction of the $\beta(n)$ estimate
- $\mathbf{K}$ is called the correction gain
- $\Big(\tilde{y}(n+1) - \hat{y}(n+1)\Big)$ is the innovation (or prediction error), measurement minus the prediction.

Recursive Least Squares

Lecture 15

In summary:

$$\boldsymbol{\beta}(n+1) = \boldsymbol{\beta}(n) + \frac{\mathbf{P}_x(n)\mathbf{f}(n+1)}{1+\mathbf{f}^T(n+1)\mathbf{P}_x(n)\mathbf{f}(n+1)}\left(\tilde{y}(n+1) - \mathbf{f}^T(n+1)\boldsymbol{\beta}(p)\right)$$

$$\mathbf{P}_x(n+1) = \mathbf{P}_x(n) - \frac{\mathbf{P}_x(n)\mathbf{f}(n+1)\mathbf{f}^T(n+1)\mathbf{P}_x(n)}{1+\mathbf{f}^T(n+1)\mathbf{P}_x(n)\mathbf{f}(n+1)},$$

for any $n \geq 0$, with initializers $\boldsymbol{\beta}(0)$ arbitrary, and $\mathbf{P}_x(0)$ arbitrary but positive definite e.g. the identity matrix $\mathbf{I}$. It should/will converge to good values.

Draft

RLS with Forgetting Factor
Lecture 15

Draft

We modify the objective function, such that we introduce exponential forgetting with a factor $\alpha \in (0, 1]$:

$$J(\beta) = \sum_{k=1}^{n} \alpha^{n-k} \left(\tilde{y}_k - \sum_{i=1}^{p} f_i(\mathbf{X}_k) \beta_i \right)^2$$

- Newer measurements have more impact on the estimation than old ones.
- Use large α for slowly changing parameters, and small α for rapidly changing parameters.

RLS with Forgetting Factor
Lecture 15

Draft

The optimization of the redefined objective function leads to the following recursive relations:

$$\boxed{\begin{aligned}\beta(n+1) &= \beta(n) + \frac{\mathbf{P}_x(n)\mathbf{f}(n+1)}{\alpha + \mathbf{f}^T(n+1)\mathbf{P}_x(n)\mathbf{f}(n+1)}\left(\tilde{y}(n+1) - \mathbf{f}^T(n+1)\beta(n)\right)\\ \mathbf{P}_x(n+1) &= \frac{1}{\alpha}\left[\mathbf{P}_x(n) - \frac{\mathbf{P}_x(n)\mathbf{f}(n+1)\mathbf{f}^T(n+1)\mathbf{P}_x(n)}{\alpha + \mathbf{f}^T(n+1)\mathbf{P}_x(n)\mathbf{f}(n+1)}\right],\end{aligned}}$$

- However, a drawback that needs to be addressed

RLS with Forgetting Factor

Lecture 15

Draft

- When the system enters steady-state, the matrix $\mathbf{P}_x(n)\mathbf{f}(n+1)\mathbf{f}^T(n+1)\mathbf{P}_x(n)$ tends to zero. This implies

$$\mathbf{P}_x(n) \approx \frac{1}{\alpha}\mathbf{P}_x(n-1)$$

- As $\alpha < 1$, $\frac{1}{\alpha}$ makes $\mathbf{P}_x(n)$ bigger than $\mathbf{P}_x(n-1)$. Although we reached steady state, $\mathbf{P}_x(n)$ begins to increase exponentially.
- Covariance Resetting: occasionally reset the matrix $\mathbf{P}(n)$, for example to the value $k \cdot \mathbf{I}$, $k > 0$.

Well done so far!

Draft