Lab-2: Converting a physical distance to a grid distance using least-square method

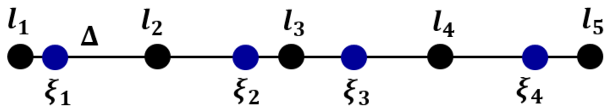
Team1: Dmitry Shadrin and Eugenii Israelit, Skoltech, 01.04.2016

Objective:

The objective of this laboratory work is to formalize a problem of placing discrete measurements on a grid scale to cover all the area of interest. This is needed for processing space data to study multi-dimensional processes. The important outcome of this exercise is the solution of problem without using classical regression models and thus avoiding bulky computations and creation of regression matrix of high dimension.

Problem:

Develop a grid scale with length of bin equal to Δ by minimizing distance between measurements ξ_i and grid nodes l_i. That means to find starting grid node l_1 and bin length Δ. The number of grid nodes is equal to N.



Solution:

Let's l_i denote grid nodes. Δ – distance between grid nodes.

$$l_{i+1} = l_i + \Delta$$

Satellite measurements $\xi_i, i = 1,2, \dots N$ are available at sequent times t_i.
But distance between measurements is variable
$\xi_i = l_i + \varepsilon_i$, ε_i – random uncorrelated unbiased noise with constant variance σ_ε^2

$$l_i = l_1 + (i-1)\Delta$$

1. Let's introduce a vector X with two components l_1 and Δ that should be found

$$X = \begin{vmatrix} l_1 \\ \Delta \end{vmatrix}$$

2. Let's introduce a matrix-row H_i

$$H_i = |1 \quad i-1|$$

$$H_i \cdot X = |1 \quad i-1| \begin{vmatrix} l_1 \\ \Delta \end{vmatrix}$$

3. Find a derivative of functional J with respect to X

$$F = \sum_{i=1}^{N} (\xi_i - H_i X)^2$$

The derivative of scalar multiple of vectors in respect to vector

$$\frac{d(YX)}{d(X)} = Y^T$$

$$F' = 2\sum_{i=1}^{N} (\xi_i - H_i X) * (-H_i)^T = \sum_{i=1}^{N} (\xi_i - H_i X) * (H_i)^T = \sum_{i=1}^{N} (H_i^T \xi_i - H_i^T H_i X)$$

And minimize the functional J the derivative should be equal to zero

$$\sum_{i=1}^{N} H_i^T \xi_i = \sum_{i=1}^{N} H_i^T H_i X$$

4. Let's introduce the following denotations

$$W = \sum_{i=1}^{N} \mathrm{H}_i^T \mathrm{H}_i$$

Where

$$\mathrm{H_i^T H_i} = \begin{vmatrix} 1 \\ \mathrm{i} - 1 \end{vmatrix} \cdot |1 \quad \mathrm{i} - 1| = \begin{vmatrix} 1 & \mathrm{i} - 1 \\ \mathrm{i} - 1 & (\mathrm{i} - 1)^2 \end{vmatrix}$$

Then

$$\mathrm{W} = \sum_{\mathrm{i}=1}^{\mathrm{N}} \begin{vmatrix} 1 & \mathrm{i} - 1 \\ \mathrm{i} - 1 & (\mathrm{i} - 1)^2 \end{vmatrix}$$

Sum of natural numbers:

$$\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{i} = \frac{\mathrm{N(N+1)}}{2}, \qquad \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{i}^2 = \frac{\mathrm{N(N+1)(2N+1)}}{6}$$

Then

$$\mathrm{W} = \begin{vmatrix} \mathrm{N} & \frac{\mathrm{(N-1)N}}{2} \\ \frac{\mathrm{(N-1)N}}{2} & \frac{\mathrm{(N-1)N(2N-1)}}{6} \end{vmatrix}$$

```
n = length(KSI);

W = [ n          (n-1)*n/2           ;
     (n-1)*n/2   (n-1)*n*(2*n-1)/6 ];
```

5. Let’s introduce the following denotations

$$C = \sum_{i=1}^{N} H_i^T \xi_i$$

```
C = [ sum(KSI);
      sum(KSI.*((1:n) - 1)) ];
```

6. Determine unknown vector X from Equation (7)

$$C = W \cdot X$$

Let’s multiply both parts on A^{-1}

$$W^{-1}C = W^{-1}W \cdot X$$

Then

$$W^{-1}C = X$$

$$X = W^{-1}C$$

```
X = W \ C;
```

Determine the covariance matrix of estimation error of vector X

$$cov(X) = \sigma_\varepsilon^2 W^{-1}$$

$$\sigma_\varepsilon^2 = \frac{1}{N-2} \sum_{i-1}^{N} (\xi_i - l_i)^2$$

```
L = X(1)+(0:n-1)*X(2);
varX = sum((KSI - L).^2) / (n - 2);
covX = varX .* inv(W);
```

Appendix 1: MatLab Source code:

```
%% Lab-2: Converting a physical distance to a grid distance using least-square method
% Team1: Dmitry Shadrin and Eugenii Israelit, Skoltech, 01.04.2016 v1

%% Prepare Data
clc; clear; close all;
KSI = importdata('ksi2_N_100_sigma_7_L1_1000_delta_10.mat');
% KSI = importdata('ksi1_N_100_000_sigma_7_L1_1000_delta_10.mat');

%% Determine unknown vector X
n = length(KSI);

W = [ n          (n-1)*n/2          ;
     (n-1)*n/2   (n-1)*n*(2*n-1)/6 ];

C = [ sum(KSI);
      sum(KSI.*((1:n) - 1)) ];

X = W \ C;

%% Determine the covariance matrix of estimation error of vector X
L = X(1)+(0:n-1)*X(2);
varX = sum((KSI - L).^2) / (n - 2);
covX = varX .* inv(W);
```