

Exercise 3: Linear Least Squares and Confidence Ellipsoids
(to be returned on Nov 23, 2018, 10:00 in SR 00-010/014,
or before in building 102, 1st floor, 'Anbau')

Prof. Dr. Moritz Diehl, Tobias Schöls, Katrin Baumgärtner, Alexander Petrov

The aim of this sheet is to strengthen your knowledge in least squares estimation and introduce to covariance ellipsoids.

Exercise Tasks

1. ON PAPER: Given the function $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ with $f(x) = x^\top Q x + c^\top x$ and fixed $c \in \mathbb{R}^n$.
 - (a) Consider the not necessarily symmetric matrix $Q \in \mathbb{R}^{n \times n}$ and compute the gradient $\nabla f(x) \in \mathbb{R}^n$ and the Hessian $\nabla^2 f(x) \in \mathbb{R}^{n \times n}$ of this function for any x . (2 points)
 - (b) If Q is symmetric, what properties does it have to fulfil such that the unique minimizer x^* can be computed? (1 point)
 - (c) Compute the unique minimizer and the minimum function value $f(x^*)$ under the correct assumptions. (1 point)

Hint: You can re-write a matrix-vector product $b = Qx$ as $b_i = \sum_{j=1}^n Q_{ij} x_j$, for $i = 1, \dots, m$.

2. ON PAPER: In the lecture notes, the sample variance S^2 is defined as

$$S^2 = \frac{1}{N-1} \sum_{n=1}^N (Y(n) - M(Y(N)))^2,$$

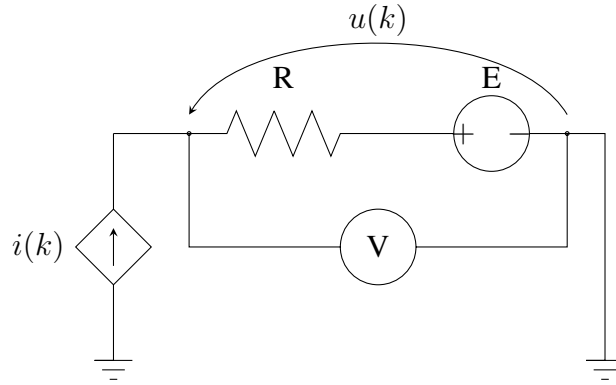
where $M(Y(N))$ is the sample mean (see lecture notes ch. 2.4 p. 17). Explain, why the division by $N-1$ is preferable over N . (2 points)

Hint: Calculate the expected value of the sample variance and compare it to the expected value of the mean squared deviations estimator.

3. MATLAB: Implement a function called `myVar(vec)` that computes the sample variance for a given vector `vec`. Use the formula from the previous exercise task to implement the variance. (1 point)

Hint: You can use MATLAB's `var` command to check your implementation.

4. Consider the following experimental set up to estimates the values of E and R .



The experiment consists of N measurements of the voltage $u(k)$ for different values of $i(k)$. We can assume that the input variable $i(k)$ is not affected by noise, but that the measurements $u(k)$ are affected by i.i.d. additive noise $n_u(k)$ and model this by:

$$u(k) = m(k) + n_u(k) \text{ where } m(k) = E + R \cdot i(k).$$

Tasks:

(9 points)

- MATLAB: Load the datasets provided on the website containing the measurements into MATLAB. Plot each dataset in a corresponding plot using the `subplot` command.
 - MATLAB: Use a Least Squares estimator in matrix form to find the experimental values of R , E for each dataset. Plot the linear fit through the measurement data.
 - MATLAB: Plot a histogram of the residuals defined as $r(k) = m(k) - u(k)$, where $m(k)$ is the voltage determined using the model, your LS-estimates and the values of $i(k)$; $u(k)$ are the measurements.
 - PAPER: Which dataset is noisier? Give an educated guess of the type of noise.
 - MATLAB: Estimate the variance of the measurement noise for each dataset.
 - MATLAB: Compute an estimate of the covariance matrix $\hat{\Sigma}$ for your earlier calculated LS-estimators for each data set.
- Hint: Check section 4.7 on the script.*
- MATLAB: The code below uses the covariance Σ to extract the parameters needed to plot a confidence ellipsoid associated with E and R ; more specifically, the 1σ confidence ellipsoid. The code example assumes $\hat{\theta} = [E, R]$, if your estimate vector considers the parameters in different order, your indexing will have to change. Make sure you understand how the confidence ellipsoids are calculated. Plot and compare both confidence ellipsoids on the same plot.
 - PAPER: Can you explain why the ellipsoids are different, despite using the same model for both measurements and similar values for the estimates?

```

1 %Compute the eigenvalues and eigenvectors for estimators
2 [V1, D1] = eig(sigma_est1); %eig_values and eig_vectors
3 [V2, D2] = eig(sigma_est2); %eig_values and eig_vectors
4
5 %Generate coordinates of 50 points on a unit circle
6 xy = [cos(linspace(0, 2*pi, 50)); sin(linspace(0, 2*pi, 50))];
7
8 %Generate the points of the confidence ellipse
9 xy_ellipse1 = [E_est1; R_est1]*ones(1,50) + V1*sqrt(D1)*xy;
10 xy_ellipse2 = [E_est2; R_est2]*ones(1,50) + V2*sqrt(D2)*xy;

```

This sheet gives 14 points in total.