Computational Intelligence SS20
Homework 1
Maximum Likelihood Estimation

Harald Leisenberger

Tutor:	Ismail Geles, ismail.geles@student.tugraz.at
Points to achieve:	20 pts
Extra points:	5* pts
Info hour:	Yet to be announced!
Deadline:	21.04.2020 23:59
Hand-in mode:	Use the **cover sheet** from the website. Submit (i) all **python files (as *.zip)** and (ii) a colored version of **your report (as *.pdf)** at the teachcenter `https://tc.tugraz.at`. (Please name your files *hw1-Familyname1Familyname2Familyname3.pdf* and *hw1-Familyname1Familyname2Familyname3.zip*)

General remarks

Your report must be self-contained and must therefore include all relevant plots, results, and discussions. Your submission will be graded based on:

- The correctness of your results (Is your code doing what it should be doing? Are your plots consistent with what algorithm XY should produce for the given task? Is your derivation of formula XY correct?)
- The depth and correctness of your interpretations (Keep your interpretations as short as possible, but as long as necessary to convey your ideas)
- The quality of your plots (Is everything important clearly visible, are axes labeled, . . . ?)

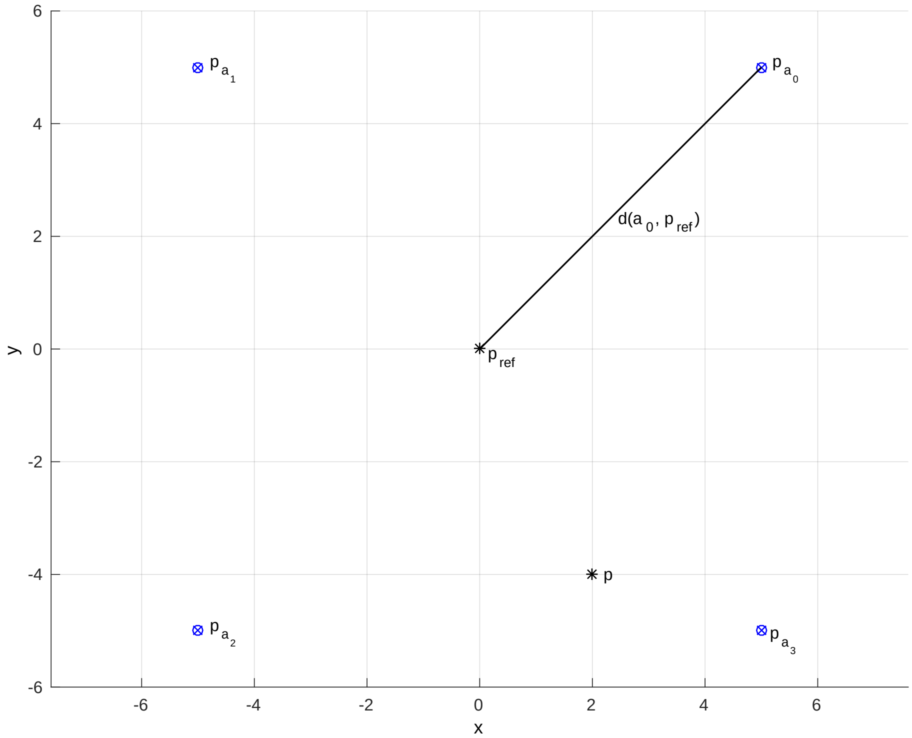

Figure 1: Four anchors (at known positions p_{a_i}, like the satellites in GPS) should be used to estimate the position of an agent, indicated as $\mathbf{p}$. For calibrating the anchors we use $\mathbf{p}_{\mathrm{ref}}$.

1 Introduction

Download `ML.zip` from the course website and unzip. `HW1_skeleton.py` contains a skeleton of the main script that you should implement.

The aim of this assignment is to estimate the position $\mathbf{p} = [x, y]^T$ of an *agent* based on (noisy) distance measurements from $N_A = 4$ *anchors* that are denoted by a_i where $i = 0, \ldots, N_A - 1$. These anchors are shown in Fig 1, and are at the known positions

$$\mathbf{p}_{\mathrm{a}_0} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}, \quad \mathbf{p}_{\mathrm{a}_1} = \begin{bmatrix} -5 \\ 5 \end{bmatrix}, \quad \mathbf{p}_{\mathrm{a}_2} = \begin{bmatrix} -5 \\ -5 \end{bmatrix}, \quad \mathbf{p}_{\mathrm{a}_3} = \begin{bmatrix} 5 \\ -5 \end{bmatrix}. \tag{1}$$

2 Maximum Likelihood Estimation of Model Parameters [6 Points]

Because of measurement errors we do not know the *true* distance $d(a_i, \mathbf{p})$ between the anchor a_i and the agent but only have access to N noisy distance measurements $\tilde{d}_n(a_i, \mathbf{p})$. In order to correct for these errors, we need a statistical model that describes the error in the measurements and relates it to the unknown parameter $\mathbf{p}$ that we intent to estimate.

First, we must specify the statistical model for each anchor and estimate the according model parameters. In order to do so we performed $N = 2000$ reference measurements to an agent at known position $\mathbf{p}_{\mathrm{ref}}$ for each anchor; all N measurements form the vector $\tilde{\mathbf{d}}(a_i, \mathbf{p})$. These measurements are contained in the file `reference_i.data` as an $(N \times N_{\mathrm{A}})$ matrix. Note that the dependence between distance and the position is given as the Euclidean distance between an anchor and the agent, i.e.,

$$d(a_i, \mathbf{p}) = \sqrt{(x_i - x)^2 + (y_i - y)^2}. \tag{2}$$

The distance measurements to the i-th anchor are distributed according to one of the following two distributions:

$$\text{Case I (Gaussian):} \quad p\big(\tilde{d}_n(a_i,\mathbf{p})|\mathbf{p}\big) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{\left(\tilde{d}_n(a_i,\mathbf{p})-d(a_i,\mathbf{p})\right)^2}{2\sigma_i^2}} \tag{3}$$

$$\text{Case II (Exponential):} \quad p\big(\tilde{d}_n(a_i,\mathbf{p})|\mathbf{p}\big) = \begin{cases} \lambda_i e^{-\lambda_i\left(\tilde{d}_n(a_i,\mathbf{p})-d(a_i,\mathbf{p})\right)}, & \tilde{d}_n(a_i,\mathbf{p}) \geq d(a_i,\mathbf{p})) \\ 0 & \text{else} \end{cases} \tag{4}$$

We consider three different scenarios where each scenario considers a different assignment of the measurement models to the anchors:

Scenario 1: Measurements of all anchors follow the Gaussian model.

Scenario 2: Measurements of one anchor follows the Exponential model, the other ones follow the Gaussian model.

Scenario 3: Measurements of all anchors follow the Exponential model.

For all scenarios we will estimate the model parameters θ (λ_i or σ_i depending on the model) using the maximum likelihood method, i.e.,

$$\theta_{ML} = \underset{\theta}{\operatorname{argmax}}\, p\big(\tilde{\mathbf{d}}(a_i,\mathbf{p})|\mathbf{p}\big) \tag{5}$$

$$= \underset{\theta}{\operatorname{argmax}} \prod_{n=0}^{N-1} p\big(\tilde{d}_n(a_i,\mathbf{p})|\mathbf{p}\big). \tag{6}$$

Therefore, you have to perform the following three steps:

- For scenario 2, find out which anchor has exponentially distributed measurements.
- Analytically derive the maximum likelihood solution for the exponential distribution.
- Estimate the parameters for all anchors, i.e., estimate σ_i^2 and λ_i in all 3 scenarios using the maximum likelihood method according to (6).

3 Estimation of the Position

After specification of the statistical model we proceed to the position estimation of the agent at an unknown position $\mathbf{p}$. Note that the estimation of the position must not take into account the true position $\mathbf{p}_{\text{true}}$ but only the measurement data provided in `measurements_i.data` . The true position $\mathbf{p}_{\text{true}}$ is only provided for error evaluation.

Ideally we would like to obtain the ML estimation of the position $\mathbf{p}$ for the n-th measurement of all anchors, i.e.,

$$\hat{\mathbf{p}}_{ML}(n) = \underset{\mathbf{p}}{\operatorname{argmax}} \prod_{i=0}^{N_{\text{A}}-1} p\big(\tilde{d}_n(a_i,\mathbf{p})|\mathbf{p}\big), \tag{7}$$

$$= \underset{\mathbf{p}}{\operatorname{argmax}}\, p(\tilde{\mathbf{d}}_n(\mathbf{p})|\mathbf{p}) \tag{8}$$

where the vector $\tilde{\mathbf{d}}_n(\mathbf{p})$ contains all $N_{\text{A}} = 4$ distance measurements.

3.1 Least-Squares Estimation of the Position [3 Points]

The nonlinear dependence in (2), however, prevents a closed form solution of an ML estimator for $\mathbf{p}$. A popular approximation to the maximum likelihood estimator is the least-squares estimator that minimizes the sum of squares of the measurement errors:

$$\begin{aligned} \hat{\mathbf{p}}_{ML}(n) \approx \hat{\mathbf{p}}_{LS}(n) &= \underset{\mathbf{p}}{\operatorname{argmin}} \sum_{i=0}^{N_\mathrm{A}-1} \left(\tilde{d}_n(a_i, \mathbf{p}) - d(a_i, \mathbf{p})\right)^2 \\ &= \underset{\mathbf{p}}{\operatorname{argmin}} \|\tilde{\mathbf{d}}_n(\mathbf{p}) - \mathbf{d}(\mathbf{p})\|^2, \end{aligned} \tag{9}$$

where $\mathbf{d}(\mathbf{p})$ is a vector that contains the true distances to all anchors (according to (2)). Therefore, you have to perform the following task:

- Show analytically that for scenario 1 (all anchors are Gaussian), the least-squares estimator of the position is equivalent to the maximum likelihood estimator, i.e., you have to prove that (8) equals (9).

3.2 Gauss-Newton Algorithm for Position Estimation [9 Points]

Because of the nonlinear dependence between distance and position in (2), an iterative algorithm is required to obtain the least-squares approximation. We will use the Gauss-Newton algorithm: it requires linearization, i.e., the first-order derivatives, of the measurement error. The collection of all derivatives is the Jacobian matrix $\mathbf{J}(\mathbf{p})$ that has dimension $N_\mathrm{A} \times 2$, with both columns defined according to

$$[\mathbf{J}(\mathbf{p})]_{i,1} = \frac{\partial\left(\tilde{d}_n(a_i, \mathbf{p}) - d(a_i, \mathbf{p})\right)}{\partial x}, \quad [\mathbf{J}(\mathbf{p})]_{i,2} = \frac{\partial\left(\tilde{d}_n(a_i, \mathbf{p}) - d(a_i, \mathbf{p})\right)}{\partial y}. \tag{10}$$

After an initial guess $\hat{\mathbf{p}}^{(0)}$, the algorithm refines the position-estimate in the t-th iteration as

$$\hat{\mathbf{p}}^{(t+1)} = \hat{\mathbf{p}}^{(t)} - \left(\mathbf{J}^T(\hat{\mathbf{p}}^{(t)})\mathbf{J}(\hat{\mathbf{p}}^{(t)})\right)^{-1} \mathbf{J}^T(\hat{\mathbf{p}}^{(t)}) \left(\tilde{\mathbf{d}}_n(\mathbf{p}) - \mathbf{d}(\hat{\mathbf{p}}^{(t)})\right) \tag{11}$$

The algorithm stops after a previously defined maximum number of iterations or if the change in the estimated position is smaller than a chosen tolerance value γ, i.e., if $\|\hat{\mathbf{p}}^{(t)} - \hat{\mathbf{p}}^{(t-1)}\| < \gamma$.

Implement and evaluate the Gauss-Newton algorithm according the following tasks:

- Implement the Gauss-Newton algorithm to find the least-squares estimate for the position. Therefore, write a function `least_squares_GN(p_anchor,p_start,measurements_n, max_iter, tol)`, which takes the $(2 \times N_\mathrm{A})$ anchor positions, the (2×1) initial position, the $(N_\mathrm{A} \times 1)$ distance measurements, the maximum number of iterations, and the chosen tolerance as input. You may choose suitable values for the tolerance and the maximum number of iterations on your own. The output is the estimated position. [4 Points]
- Run your algorithm for each of the $N = 2000$ independent measurements; choose the initial guess $\hat{\mathbf{p}}^{(0)}$ randomly according to a uniform distribution within the square spanned by the anchor points. Take a closer look at:
  - The mean and variance of the position estimation error $\|\hat{\mathbf{p}}_\mathrm{LS} - \mathbf{p}\|$.
  - Scatter plots of the estimated positions. Fit a two-dimensional Gaussian distribution to the point cloud of estimated positions and draw its contour lines with the provided function `plot_gauss_contour(mu,cov,xmin,xmax,ymin,ymax,title)`. Do the estimated positions look Gaussian?

- Compare the different scenarios by looking at the cumulative distribution function (CDF) of the position estimation error, i.e. the probability that the error is smaller than a given value. Use the provided function `Fx,x = ecdf(realizations)` for the estimation of the CDF and `plt.plot(x,Fx)` for plotting. What can you say about the probability of large estimation errors, and how does this depend on the scenario?

Discuss the performance of your estimation based on the above analysis! [4 Points]

- Consider scenario 2: the least-squares estimator is equivalent to the maximum likelihood estimator for Gaussian error distributions. Therefore, neglect the anchor with the exponentially distributed measurements! Compare your results to before! What can you observe (Gaussianity of the estimated positions, probability of large errors, ...)? [1 Point]

3.3 Numerical Maximum Likelihood Estimation of the Position

For non-Gaussian distributed data, the maximum likelihood-estimator is in general not equivalent to the least-squares estimator. In this example, we consider scenario 3 (all anchors have exponentially distributed measurements) and compare the least-squares estimator to a maximum likelihood estimator. As a closed-form solution is not available because of the non-linearity we will evaluate the likelihood numerically and select the point that maximizes it.

3.3.1 Single Measurement [2 Points]

Let us consider the first measurement $n = 0$ for all all N_A anchors now. For this single measurement $\tilde{\mathbf{d}}_0(\mathbf{p})$ you have to perform the following tasks:

- Evaluate the joint likelihood $p\big(\tilde{\mathbf{d}}_0(\mathbf{p})|\mathbf{p}\big)$ according to (4) over a 2-dimensional grid with a resolution of 0.05. Confine the evaluation to the square region enclosed by the anchors.
- Why might it be hard to find the maximum of this function with a gradient ascent algorithm using an arbitrary starting point within the evaluation region?
- Is the maximum at the true position? Explain your answer!

3.3.2 Multiple Measurements [5* Points]

Now, estimate the position for all N measurements. For all individual measurement, compute a numerical maximum likelihood estimate based on the joint likelihood evaluated over the grid, i.e., take the maximum of $p\big(\tilde{\mathbf{d}}_n(\mathbf{p})|\mathbf{p}\big)$ as an estimate. Then perform the following tasks:

- Compare the performance of this estimator (using the same considerations as above) to the least-squares algorithm for the data from scenario 3.
- Is this comparison fair? Is this truly a maximum likelihood estimator?
- Using a Gaussian *prior* pdf $p(\mathbf{p})$ centered at $\mathbf{p}$ with a diagonal covariance matrix $\Sigma = \mathrm{diag}\{\sigma_p^2\}$ with $\sigma_p = 1$, compute a Bayesian estimator for the position

$$\hat{\mathbf{p}}_{\mathrm{Bayes}} = \operatorname*{argmax}_{\mathbf{p}} p\big(\tilde{\mathbf{d}}_0(\mathbf{p})|\mathbf{p}\big)p(\mathbf{p}). \tag{12}$$

  Again, use the evaluation over the grid. Describe the results! Does the prior knowledge enhance the accuracy?