Computational Intelligence SEW SS17
Homework 4
Maximum Likelihood Estimation

Christian Knoll

Tutor:	Florian Kaum, florian.kaum@student.tugraz.at
Points to achieve:	17 pts
Extra points:	5* pts
Info hour:	29.05.2016, 12:00-13:00, HS i11
Deadline:	02.06.2016 23:59
Hand-in mode:	Use the **cover sheet** from the website. Submit all **python files and a colored version of your report (as.pdf)** at `https://courses-igi.tugraz.at/`. (Please name your archive *hw5-Familyname1Familyname2Familyname3.zip*)
Course information:	`https://www.spsc.tugraz.at/courses/computational-intelligence-sew`
Newsgroup:	tu-graz.lv.ew

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, . . . ?)

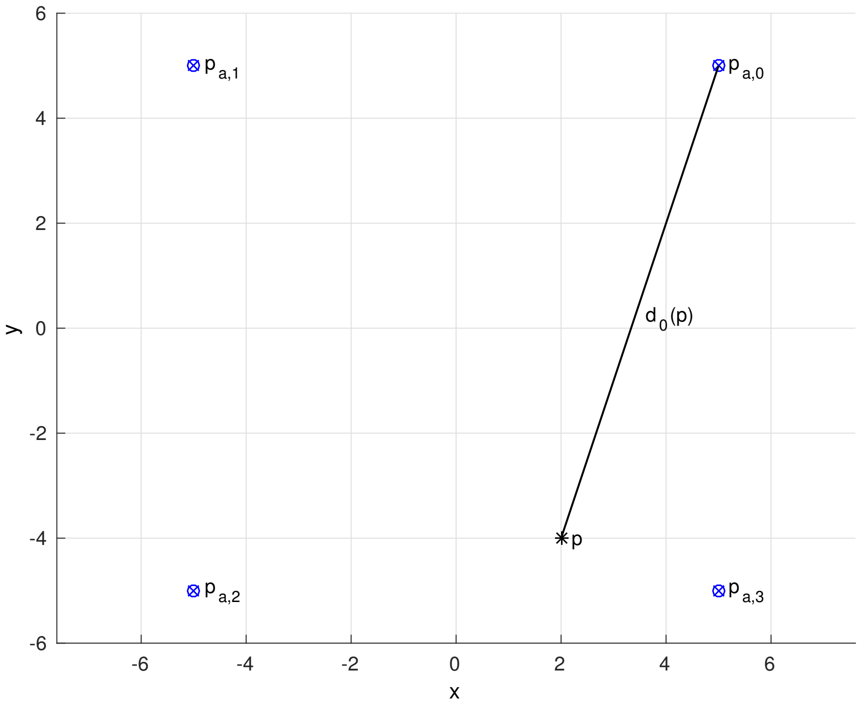

Figure 1: Four anchors (at known positions $p_{a,i}$, like the satellites in GPS) should be used to estimate the position of an agent, indicated as $\mathbf{p}$)

1 Maximum Likelihood Estimation

Download `ML.zip` from the course website and unzip. `HW5_skeleton.py` contains a skeleton of the main script that you should implement.

1.1 Scenario and model

The aim of this exercise is to estimate the position $\mathbf{p} = [x, y]^T$ of an agent using (independent and identically distributed) noisy distance measurements r_i, $i = 0, \ldots, N_\mathrm{A} - 1$ to the $N_\mathrm{A} = 4$ anchor nodes. These are shown in Fig. 1, and are at the positions

$$\mathbf{p}_{\mathrm{a},0} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}, \quad \mathbf{p}_{\mathrm{a},1} = \begin{bmatrix} -5 \\ 5 \end{bmatrix}, \quad \mathbf{p}_{\mathrm{a},2} = \begin{bmatrix} -5 \\ -5 \end{bmatrix}, \quad \mathbf{p}_{\mathrm{a},3} = \begin{bmatrix} 5 \\ -5 \end{bmatrix}. \tag{1}$$

We are not able to measure the *true* distance $d_i(\mathbf{p})$ between the agent at position $\mathbf{p}$ and the anchors. Hence, we need a statistical model that describes the error in the measurement r_i and relates it to the unknown parameter $\mathbf{p}$ that we want to estimate. We consider two cases, in which the distance measurements to the i-th anchor are distributed as

$$\text{Case I (Gaussian):} \quad p(r_i|\mathbf{p}) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(r_i - d_i(\mathbf{p}))^2}{2\sigma_i^2}} \tag{2}$$

$$\text{Case II (Exponential):} \quad p(r_i|\mathbf{p}) = \begin{cases} \lambda_i e^{-\lambda_i (r_i - d_i(\mathbf{p}))}, & r_i \geq d_i(\mathbf{p}) \\ 0 & \text{else} \end{cases} \tag{3}$$

The dependence on the parameter $\mathbf{p}$ is given as the Euclidean distance to the i-th anchor, i.e.

$$d_i(\mathbf{p}) = \sqrt{(x_i - x)^2 + (y_i - y)^2}. \tag{4}$$

The nonlinear dependence in (4) does not allow for a closed form solution of an ML estimator for $\mathbf{p}$. A popular approximation to the maximum likelihood estimator is the least-squares estimator, i.e., to minimize the sum of squares of the measurement errors:

$$\hat{\mathbf{p}}_{\mathrm{ML}} = \arg\max_{\mathbf{p}} p(\mathbf{r}|\mathbf{p}) = \arg\max_{\mathbf{p}} \prod_{i=1}^{N_{\mathrm{A}}} p(r_i|\mathbf{p}) \tag{5}$$

$$\approx \hat{\mathbf{p}}_{\mathrm{LS}} = \arg\min_{\mathbf{p}} \sum_{i=1}^{N_{\mathrm{A}}} (r_i - d_i(\mathbf{p}))^2 = \|\mathbf{r} - \mathbf{d}(\mathbf{p})\|^2 \tag{6}$$

where the vector $\mathbf{r}$ contains all $N_A = 4$ distance measurements r_i, and $\mathbf{d}(\mathbf{p})$ is a vector that contains distances to all anchors $d_i(\mathbf{p})$, according to (4).

However, d_i still has a nonlinear dependence of $\mathbf{p}$. Therefore, we will use an iterative algorithm – the Gauss-Newton algorithm – to obtain the least-squares approximation: It requires the calculation of the Jacobian matrix of the measurement errors, which collects the first-order derivatives (linearizations) of the measurement errors. In this example, this matrix has the dimensions ($N_{\mathrm{A}} \times 2$). The two columns are defined as

$$[\mathbf{J}_r(\mathbf{p})]_{i,1} = \frac{\partial(r_i - d_i(\mathbf{p}))}{\partial x}, \quad [\mathbf{J}_r(\mathbf{p})]_{i,2} = \frac{\partial(r_i - d_i(\mathbf{p}))}{\partial y}. \tag{7}$$

The algorithm starts with an initial guess $\hat{\mathbf{p}}^{(0)}$ and updates the parameter in the t-th iteration as

$$\hat{\mathbf{p}}^{(t+1)} = \hat{\mathbf{p}}^{(t)} - \left(\mathbf{J}_d^T(\hat{\mathbf{p}}^{(t)})\mathbf{J}_d(\hat{\mathbf{p}}^{(t)})\right)^{-1} \mathbf{J}_d^T(\hat{\mathbf{p}}^{(t)}) \left(\mathbf{r} - \mathbf{d}(\hat{\mathbf{p}}^{(t)})\right) \tag{8}$$

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.,$\|\hat{\mathbf{p}}^{(t)} - \hat{\mathbf{p}}^{(t-1)}\| < \gamma$.

1.2 Maximum Likelihood Estimation of Model Parameters [6 Points]

First we must specify the statistical model and estimate the according model parameters. *Hint:* You may assume that the true position – and the true distances $d_i(\mathbf{p})$ – are known

We use three different scenarios for the evaluation. Each scenario considers a different assignment of the measurement models (2) and (3):

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

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

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

The according measurements are contained in the files `HW5_x.data` as ($N \times N_{\mathrm{A}}$) matrices $\boldsymbol{R}$, where $N = 2000$ is the number of measurements.

1. For scenario 2, find out which is the anchor with exponentially distributed measurements.
2. Analytically derive the maximum likelihood solution for the exponential distribution.
3. Estimate the parameters of the measurement models (2) and (3), i.e., estimate σ_i^2 and λ_i for all anchors in all 3 scenarios using the maximum likelihood method.

1.3 Least-Squares Estimation of the Position [9+2* Points]

After specification of the statistical model we can estimate the position of the agent. Note that the estimation of the position must not take into account the true position, but only the measurement data provided.

- Implement the Gauss-Newton algorithm to find the least-squares estimate for the position. Write a function `LeastSquaresGN(p_anchor, p_start, r, 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 estimates, 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]

- For all three scenarios, evaluate your estimation algorithm using the provided data. For each of the $N = 2000$ independent measurements, choose the starting position $\mathbf{p}_0$ randomly according to a uniform distribution within the square spanned by the anchor points. Have a 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. You can use the provided function `plotGaussContour(mu,cov)`. Do the estimated positions look Gaussian?
  - You can 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 error. You can use the provided function `Fx,x = ecdf(realizations)` for the estimation of the CDF. For plotting, use `plt.plot(x,Fx)`. What can you say about the probability of large estimation errors?

  [4 Points]

- Compare the performance of scenario 2 with the case that you do not use the anchor with the exponentially distributed measurements at all! What can you observe (Gaussianity of the estimated positions, probability of large errors, ...)? [1 Point]

- Show analytically that for scenario 1 (joint likelihood for the distances is Gaussian), the least-squares estimator of the position is equivalent to the maximum likelihood estimator, i.e., (5) equals (6). [2* Points]

1.4 Numerical Maximum-Likelihood Estimation of the Position [2+3* Points]

For non-Gaussian distributed data, the maximum likelihood-estimator is in general not equivalent to the least-squares estimator. In this example, we want to compare the least-squares estimator with a direct maximization of the likelihood function for scenario 3 (all anchors have exponentially distributed measurements).

1. For the first measurement (i.e. the first N_A distance estimates), compute the joint likelihood function $p(\mathbf{r}|\mathbf{p})$ according to (3) over a two dimensional grid with a resolution of 5 cm. Confine the evaluation to the square region that is 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? [2 Points]

2. For all measurements, compute a numerical maximum likelihood estimate based on the the joint likelihood function evaluated over the grid, i.e. just take the maximum of $p(\mathbf{r}|\mathbf{p})$ as an estimate. Compare the performance of this estimator (using the same considerations as above) with the least-squares algorithm for the data from scenario 3.
   Is the comparison fair?
   Is this truly a maximum likelihood estimator?
   What happens if you consider not only a single measurement, but all 2000 measurements?
   [3* Points]