Comments on Assignment 1 (MLE):

(H. Leisenberger)

1 Introduction:

•) Besides the 4 anchors you can read from Fig. 1 the positions

of $p_{ref} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ ← known, only used for establishing the stat. models in 2

and $p = \begin{pmatrix} 2 \\ -4 \end{pmatrix}$

↑ not known! to be estimated in 3 based on the models found in 2

2 MLE of Model Parameters:

•) Notation:

-) $d(a_i, p)$ .... (euclid.) true distance between p_{a_i} and p, unknown

contained in measurements - i. data ($i = 1, 2, 3$) ↳ for dif. scenarii

-) $\tilde{d}_n(a_i, p)$ .... measured dist. betw. p_{a_i} and p for n-th measurement, noisy

-) $\underline{\tilde{d}}(a_i, p)$ .... vector of $\tilde{d}_n(a_i, p)$, $n = 1, \dots, 2000$

-) $d(a_i, p_{ref})$ .... dist. between p_{a_i} and p_{ref}, known (as we know pos. of p_{ref})

contained in reference - i. data ($i = 1,2,3$)

-) $\tilde{d}_n(a_i, p_{ref})$ .... measured dist. between p_{a_i} and p_{ref} for n-th measurement, noisy

-) $\underline{\tilde{d}}(a_i, p_{ref})$ .... vector of $\tilde{d}_n(a_i, p_{ref})$, $n = 1, \ldots, 2000$

•) On the assingnment sheet there is no proper distinction between $d(a_i, p)$ (dist. to p) and $d(a_i, p_{ref})$ (dist. to p_{ref}); the "ref" is omitted most of the time (for simplicity).

But: in section 2 we are actually dealing with $d(a_i, p_{ref})$ and the corresponding measurements, while in section 3 $d(a_i, p)$ is correctly used for what it is defined to be.

•) ad Case 1 (Gaussian distr.):

in terms of the notation used in Practical 3, we could denote $p(\tilde{d}_n(a_i, p) | p)$ as $p(\tilde{d}_n(a_i, p) | d(a_i, p), \sigma_i^2)$, ... 2

... where $d(a_i, p)$ is the mean μ of the distr. and σ^2 its variance. Note that, at least in 2 the mean $d(a_i, p)$ $(= d(a_i, p_{ref})$ in this section) is known, while σ_i^2 has to be estimated from the data.

•) ad Case 2 (Exponential distr.): this is in fact a "translated" exp. distr. whose origin is shifted by $d(a_i, p)$ to the right on the x-axis (compare to the "original" formula in pract. 3!). For $p(\tilde{d}_n(a_i, p) | p)$ we could also write $p(\tilde{d}_n(a_i, p) | \lambda_i)$.

•) $p = \binom{x}{y}$ is a parameter which occurs in both distr. implicitely in $d(a_i, p) = \sqrt{(x_i - x)^2 + (y_i - y)^2}$ and can not be estimated analytically (due to nonlinear dependence)!

•) Once again: goal in this section is only the MLE-estimation of σ_i^2 (Case 1) and λ_i (Case 2) for all dif. scenarii only using the reference data! For est. of p, we will use then the act. measurements.

3 Estimation of the position:

•) From now on, we "forget" about the ref. data & position and only take into account the measured dist. to the (unknown) p. Note that we do not longer have knowledge on the mean of the Gaussian, as we do not know the true distances $d(a_i, p)$. Of course $d(a_i, p)$ could easily be estimated by the mean of the $d_n(a_i, p)$. But this would not be helpful, as we wouldn't gain useful information about p.

•) ad Gauss-Newton alg. (3.2):

We are faced with a nonlinear least squares problem of the form

$$\| \underline{y} - \underline{f}(\underline{x}) \|^2 \xrightarrow[x]{} \min .$$

$\hookrightarrow$ nonlin. in x

That's why we can not apply a simple least squares method for lin. problems of the form $\| \underline{y} - A\underline{x} \|^2$, but have to resort to the iterative Gauss-Newton alg.. Basically it approximates the nonlin. problems by a sequence of lin. ones which it solves iteratively.

Note that you have to calculate the entries of the Jacobian in (10) analytically.

Recall: for $f : \mathbb{R}^n \to \mathbb{R}^m$, $f(x_1,\dots,x_n) = \begin{pmatrix} f_1(x_1,\dots,x_n) \\ \vdots \\ f_m(x_1,\dots,x_n) \end{pmatrix}$

the Jacobian $Jf(x_1,\dots,x_n)$ is defined to be

$$Jf(x_1,\dots,x_n) := \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x_1,\dots,x_n) & \cdots & \frac{\partial f_1}{\partial x_n}(x_1,\dots,x_n) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(x_1,\dots,x_m) & \cdots & \frac{\partial f_m}{\partial x_n}(x_1,\dots,x_n) \end{pmatrix}.$$

- For fitting a Gaussian distr. you can use scipy.stats.multivariate_normal (see also Bivariate_Gaussian.ipynb from practical 3).

- ad Numerical MLE (3.3):

  Here the ML fct. has to be evaluated only for a finite number of points on an equally spaced grid. The numerical maximizer is then the grid point with the highest value of the function.

- With gradient ascent we mean moving in the direction of the gradient (and not in the neg. direction as we do in gradient descent).